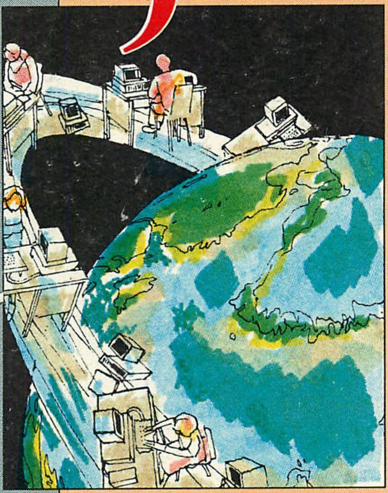
SEPTEMBER-OCTOBER 1983

VOL. 1, NO. 2 $3.95

FOR IBM PERSONAL COMPUTER USERS

TECH PC JOURNAL



IBM LOCAL AREA NETWORKS
A system overview of PCnet, including sample configurations

18 PROGRAM EDITORS
A comparative review

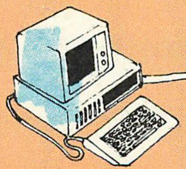
EPSON PRINTS AGAIN
The RX-80 and FX-80, plus a programmer's guide to Epson Printers

FORTRAN PITFALLS
Pitfalls revealed, bugs explained

8087 NUMBER CRUNCHING
How the Math Co-Processor stacks up: benefits and benchmarks

COBOL AND THE SERIAL PORT
Nuances of low-level access from a high-level language

WORD PROCESSOR FILE CONVERSION
Getting from one word processor to another: A program that does it



EXE AND COM FILES
DOS Executable files explained

0 14024 14357 10

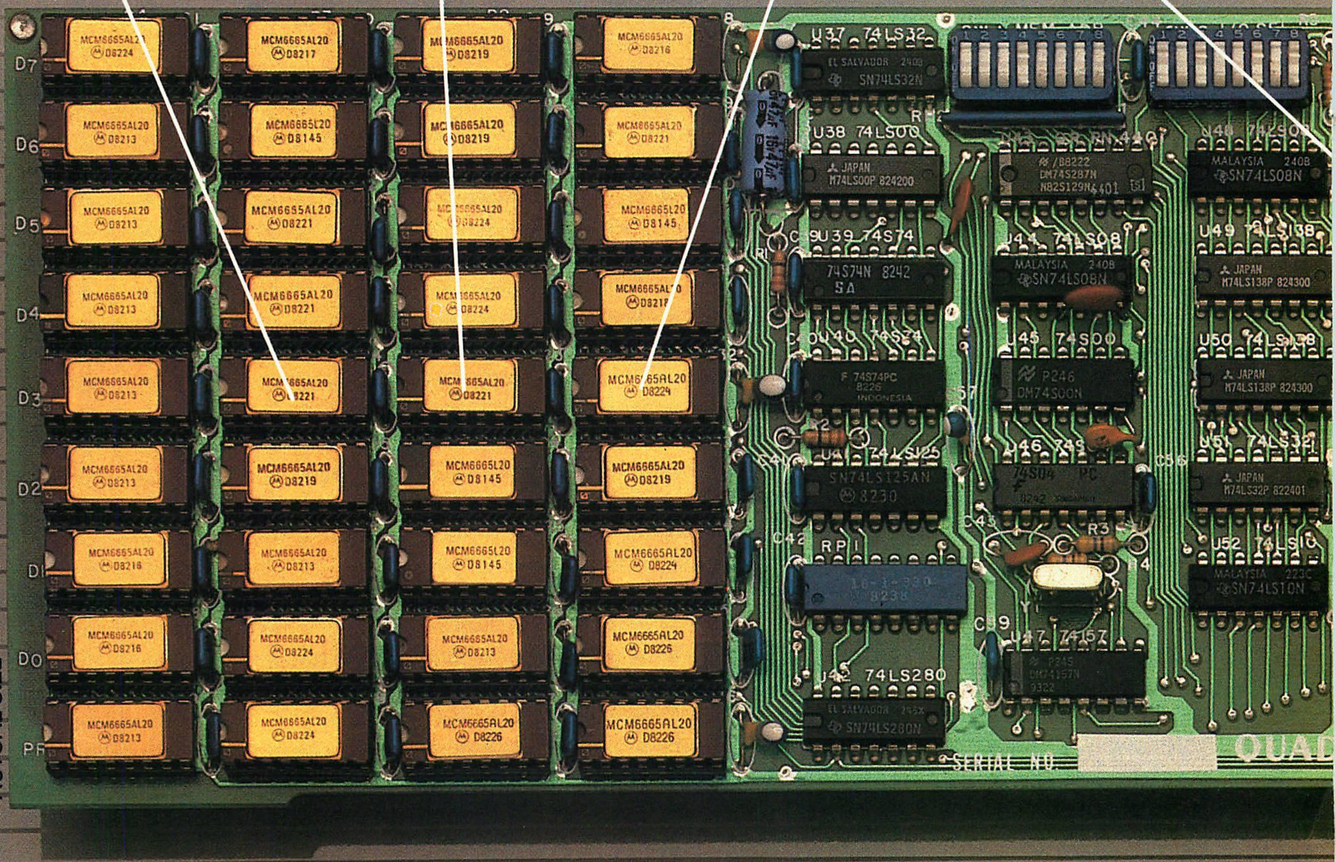
QUADBOARD™

256K Memory Expansion
Expandable in 64K increments, Quadboard™ is socketed for 256K bytes of memory. Full parity and checking standard.

Qspool™
With Qspool, you can also use part of system memory as a software print buffer. Choose from 8K to 56K of memory and stop waiting for your printer.

QuadRAM Drive™
This software program lets you use part of your total system memory as multiple disk drives. Super-fast drives that let you store important information for easy access.

Chronograph
Quadboard features a real time chronograph to always keep your system's clock/calendar up-to-date. Even when your system's been off for months.



ACTUAL SIZE

THE FIRST AND ONLY BOARD YOUR IBM PC MAY EVER NEED.

Your IBM Personal Computer is very versatile. New functions and applications are being developed for it everyday. And now with Quadram's Quadboard, you can keep your PC options open for tomorrow's technology. In the tradition of Quadram Quality, six of the most needed PC functions have been combined into one package, using only one expansion slot. Your remaining IBM slots are left free for future expansion needs.

All-On-One-Board and Software, too.
Every board you may ever need for your IBM PC is in the Quadboard package. From serial port to print buffer. Plus, there's QuadMaster software, too. Included on this disk are utilites for accurately setting the Quadboard's chronograph for time and date, and for performing diagnostics on all Quadboard functions. There's

QUADRAM QUALITY

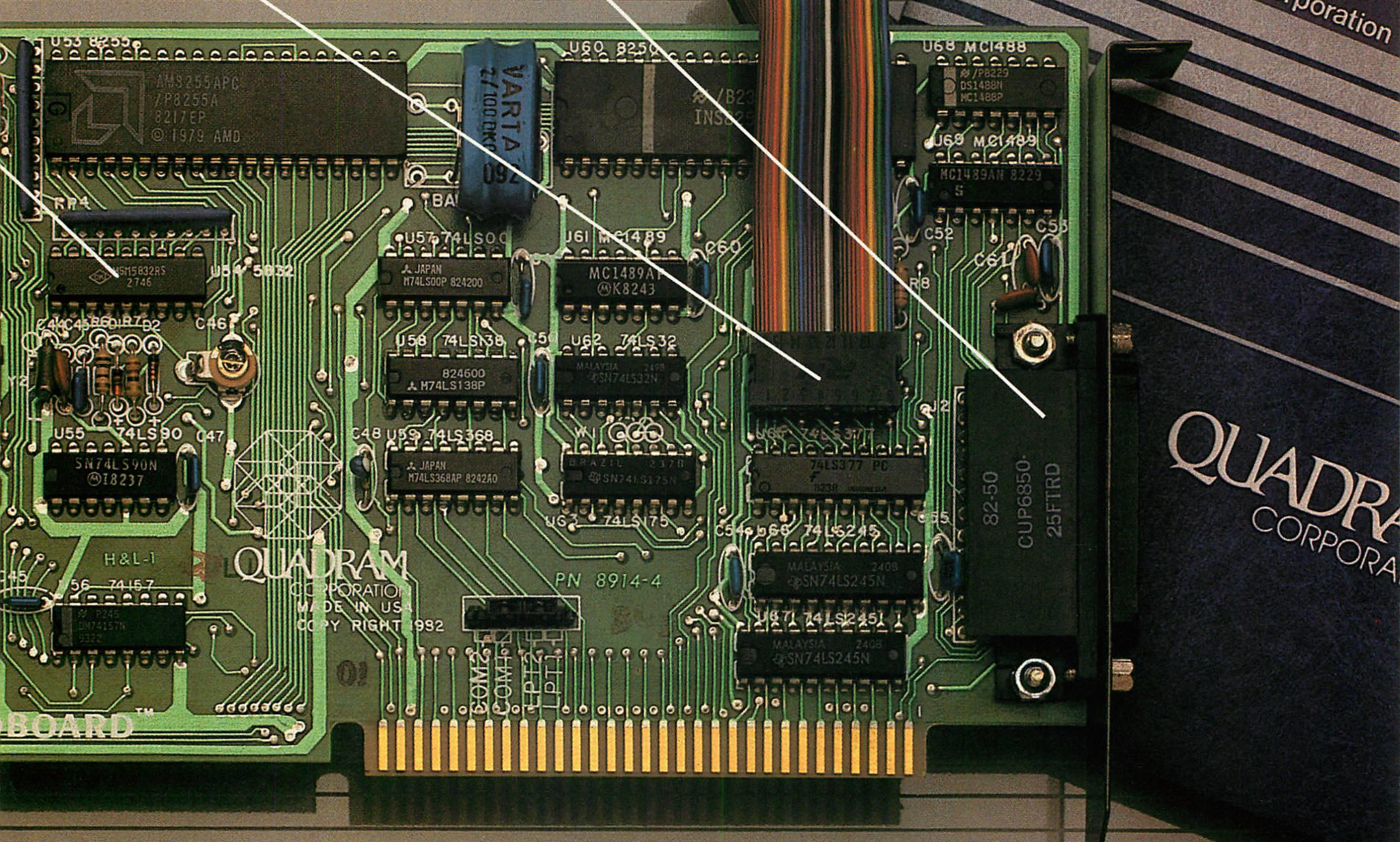
BY QUADRAM™

Parallel Port
There's a parallel I/O designed to operate most printers and parallel devices.

Or a second serial port on Quadboard II.

Serial Port
Used for connecting modems, printers, and other serial devices, Quadboard's serial adapter can be configured as COM1 or COM2 and fully supports IBM PC communications software.

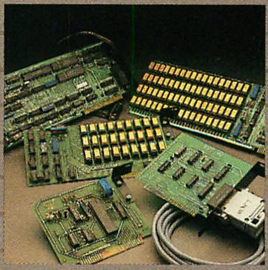
\$395 (64K)
\$595 (256K)



also Qswap,™ the feature that lets you switch line printers one and two, back and forth, as your printing needs change.

Proven Design
Quadboard is the number one IBM PC option board on the market today. Nothing even comes close, because Quadboard is designed with performance in mind. Engineered for dependability and built with only the finest components available. Each board has been thoroughly tested and "burned-in" for years of reliable service.

INCREDIBLE PRICE, ASK YOUR DEALER.
Our full line of IBM PC accessories: Quadboard II, Quad 512+, Quadcolor, Quadchrome, Quadscreen, Quadmodem, Memory Board, Serial Board (Single or Dual ports), Parallel Board, Chronograph, and Microfazer.



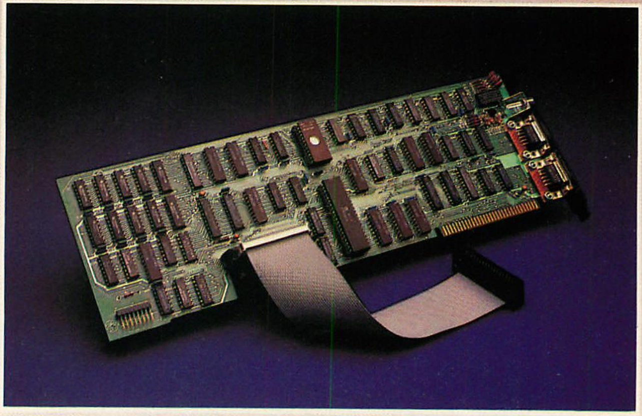
All products are sold through local personal computer dealers. If yours does not stock Quadram, please ask him to call us at (404) 923-6666.

QUADRAM CORPORATION
An Intelligent Systems Company

4355 International Blvd./Norcross, Ga. 30093
(404) 923-6666/TWX 810-766-4915 (QUADRAM NCRS)

CIRCLE NO. 191 ON READER SERVICE CARD

From Quadram comes Quadlink.™ The Apple-IBM link. The door to a whole new world of software for your Personal Computer.



Quadlink. Simply plug it inside your PC, press a few keys, and instantly your computer becomes an Apple-compatible system. Able to run virtually any Apple software package available.

Now you can choose from the largest software library ever written. Business, educational, and entertainment packages. And never worry about "compatibility" again. Quadlink makes it possible.

So enhance your PC system. Link up to a whole new world of Apple software. With Quadlink by Quadram. (And remember, Quadlink works great with Quadram's other popular PC enhancement board, the six-function Quadboard.) Available now at authorized full-service Quadram dealers worldwide.

The 'link is finally here.

QUADRAM QUALITY

QUADRAM CORPORATION
An Intelligent Systems Company

4355 International Blvd./Norcross, Ga. 30093
(404) 923-6666/TWX 810-766-4915 (QUADRAM NCRS)

With all the clamor about personal computers, a fundamental fact is often overlooked: some simply *work* better than others.

Consider the COMPAQ Portable.

A computer will make you more productive. A computer will make you more efficient. You hear it everywhere. But you don't hear much about which computer actually *works* best.

The COMPAQ Portable fits under a standard airline seat for business trips.

Works in more places

With the COMPAQ™ Portable, you can be as productive in your hotel room or your lake house as in your own office. It's a reliable companion on a business trip. It's a powerful sales aid in your customer's office.

You can move it from office to office to share its resources. You can move it into the conference room to answer questions.

What's more productive than a computer? A computer that works for you in more places.

Works with the greatest number of programs

The most important consideration when choosing a computer is "what programs will it run?"

The COMPAQ Portable runs more programs than any other portable. In fact, it runs more than most non-portables because it runs all the popular programs written for the IBM® Personal Computer without any modification. There are hundreds of them. They are available at computer stores all over the country.

Imagine the power of a *portable* word processor. There are dozens of word processing programs available for the COMPAQ Portable.

Planning, problem-solving, and "what-ifs" are a cinch with a variety of popular electronic spreadsheet programs. The COMPAQ Portable runs them all.

There are accounting programs for anything from computerizing your family budget to full-scale professional management of payables, receivables, inventory, and payroll.

There are programs for making charts and programs for communicating with other computers.

So you get portability and you don't give up problem-solving power. The combination adds up to the most useful personal computer on the market today.

Add-on options make it work the way you work

Inside the COMPAQ Portable are three open slots. Most portables don't have any. Electronic devices called expansion boards fit those slots and give the COMPAQ Portable new powers. As with programs, expansion boards designed for the IBM will work. With them, you can make your personal computer more personal.

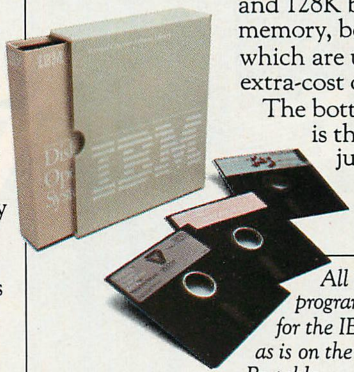
Want to check a stock price? One expansion board enables the COMPAQ Portable to handle those communications over ordinary phone lines.

Want to use your company's central computer files while you're on a trip? There are boards that allow the COMPAQ Portable to communicate with a variety of large computers.

Other boards let you hook up controllers for computer games, increase memory capacity, or connect several personal computers in a network.

The added usefulness is free

The COMPAQ Portable doesn't cost any more than an ordinary desktop computer. In fact, it costs hundreds less than a comparably equipped IBM or Apple® III. The COMPAQ Portable comes standard with one disk drive and 128K bytes of memory, both of which are usually extra-cost options. The bottom line is this—you just can't buy a more practical, useful, productive computer. Compare the COMPAQ Portable.



All the popular programs written for the IBM PC run as is on the COMPAQ Portable.

For the location of the Authorized Dealer nearest you, call 1-800-231-9966.

CIRCLE NO. 125 ON READER SERVICE CARD

COMPAQ™

The most computer you can carry

PROFESSIONAL

PROGRAMMING TOOLS

■ ProEdit II

Speed up your program development! With ProEdit II you can perform complex text manipulations with the speed and power your IBM PC was meant for— at half the price of comparable editors. Compare these features: True full screen editing with horizontal and vertical scrolling ■ Word wrap ■ Automatic disk buffering ■ Easy recovery from disk full errors ■ Full DOS command support ■ High quality documentation ■ COMMANDS that allow you to: Rapidly move, copy, replace, and delete text ■ Display and edit multiple files ■ Customize your ALT keys for use with any programming language ■ Easily repeat a set of commands with one key ■ Customize your print output ■ Custom tailor your function keys ...

These are just some of the powerful features of ProEdit II. Compare us with the competition. You will see that ProEdit II contains the features you need at an affordable price—only **$129.00**, or order just the manual for $40.00.

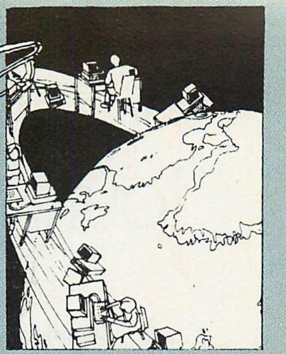
■ DYNAMIC SCREENFORMS

Dynamic Screenforms is a versatile screen design/data handling package designed for the serious microcomputer programmer, and yet so simple to use that even an amateur can create BASIC code for applications software in a fraction of the normal time. You can now create and maintain programs written in BASIC with the power and flexibility that professional programmers demand. Your programs will allow the end-user maximum full-screen data input flexibility; fast file access, and error checking. You get all of the power of Dynamic Screenforms now for only **$99.00**, or order just the manual for only $40.00.

To order direct call: 1-800-821-8088

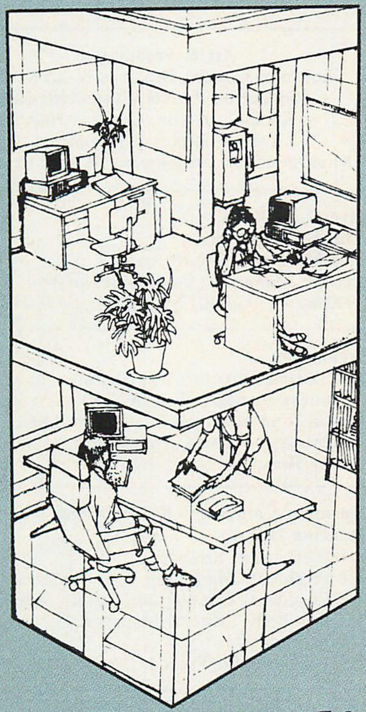
ORION SOFTWARE

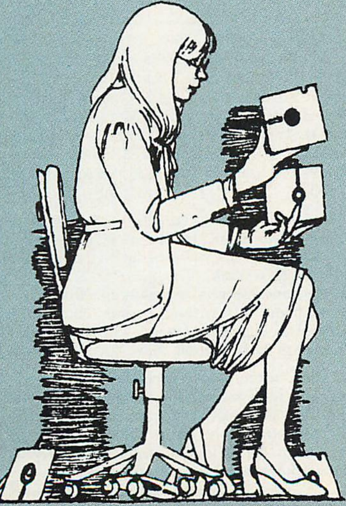
P.O. BOX 2488 • AUBURN • AL • 36831



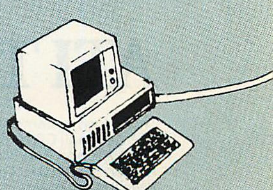
TECH JOURNAL

ARTICLES





DEPARTMENTS



TECH JOURNAL

VOL. 1, NO. 2

Editorial

Editor: Will Fastie
Managing Editor: Marjory Spraycar
Techincal Editor: Susan Glinert-Cole
Editorial Assistant: Arthur A. Gleckler
Contributing Editors: Ray Duncan, Richard Foard, Thomas V. Hoffmann, Max Stul Oppenheimer
Editorial Secretary: Diana L. Carey
Proofreading: Judith L. Kreiner, Charlene Trahan

Art & Production

Editorial Design: Douglas Ward Kelley
Production Artists: Susan Huyser, Sheri Goldstein, Cynthia Hier, Kaethe Cherney, Tom Selick
Illustrators: David Johnson, Robert Conrad

Advertising Sales

Publisher: Jeff Weiner
Sales Manager: Pauline Scherer
Account Managers: Caroline Anderson, Theodore Bahr III, Mike Mahana, Ian Smith

Circulation

Subscription Director: Chet Klimuszko, New Business Manager: Eric A. Bernhard, Renewal and Billing Manager: Shane Boel, Direct Retail Sales: Lynn Kujawa, Telemarketing Manager: Linda Rossiter

Consumer Computers & Electronics Magazine Division

President: Larry Sporn
Vice President, Marketing: J. Scott Briggs
Vice President, Circulation: Carole Mandel
Vice President, General Manager: Eileen G. Markowitz
Creative Director: Peter J. Blank

Ziff-Davis Publishing

President, Richard P. Friese; President, Consumer Magazine Division, Albert S. Traina; Executive Vice President, Marketing and Circulation, Paul H. Chook Senior Vice President, Phillip T. Heffernan Senior Vice President, Sidney Holtz Senior Vice President, Edward D. Muhlfeld Senior Vice President, Philip Sine Vice President, Baird Davis Vice President, George Morrissey Treasurer, Selwyn Taubman Secretary, Bertram A. Abrams

PC Tech Journal (ISSN 0738-0194) is published bi-monthly for $24.97 one year, $43.97 for two years. Additional postage $6.00 for Canada & Foreign. PC Tech Corp. is a subsidiary of Ziff-Davis Publishing Company, The World Trade Center, Suite 211, Baltimore, Maryland, 21202. Application to mail at Second-Class Postage Rates is Pending at New York, NY 10001. POSTMASTER: Send address changes to *PC Tech Journal*, 39 East Hanover Ave., Morris Plains, NJ 07950.

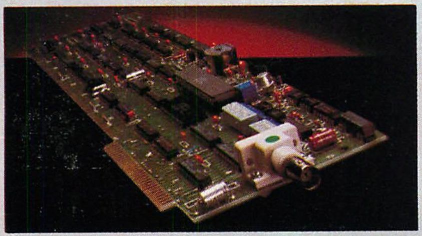
Editorial and Business Offices: *PC Tech Journal*, The World Trade Center, Suite 211, Baltimore, Maryland 21202. Editorial (301) 576-0770, Advertising PC Tech Journal, One Park Avenue, New York, N.Y. 10016 (212) 725-7947.

Share and share alike.

With SCS PCnet™... a true, low-cost network for IBM Personal Computers.

Share and grow with PCnet from Santa Clara Systems . . . the true local area network for the IBM Personal Computer (PC) and its MS-DOS operating System. PCnet allows the sharing of hard disks, floppy disks, letter-quality printers, modems, and internal memory by all networked PCs. So you can add new dimensions of capability to your system while reducing your per station costs!

True network news
With its proprietary, on-board intelligence, PCnet permits direct communication between all PCs in your network. If one goes down, the others don't. Unlike other network products, a dedicated disk server is not necessary, saving you money. In fact, no extra hardware is required to add more stations; just one PCnet board for each and you're set. Plus, PCnet is transparent to all IBM and compatible software.



Service for thousands
You have plenty of room to grow. And, as you add more users, your peripheral resource power and memory capacity increase linearly. For all that it offers, PCnet is surprisingly easy to install. Just plug a PCnet board into an I/O slot of each IBM PC you want to include in your network. Then tap into the standard coaxial cable—offered in lengths of more than one mile with PCnet!

The extras you'll like
You'll appreciate the specialized network utilities available with PCnet. Like print spooling, multi-tasking, and file-locking. Plus the ability to communicate with IBM PC look-alikes. In addition, the Pascal Bubble™ is supported under MS DOS.

Full customer support
Only Santa Clara Systems offers you worldwide support with PCnet. A full 90-day warranty. *And a complete line of hard-disk storage products tailor made for use with the PCnet,* featuring up to a ½ gigabyte of storage per PC and removable Winchester cartridges for backup.

See the difference sharing can make in your business. Our starter kit has everything you need to network two PCs right away—two PCnet boards, cable, and documentation. For more information, call or write us today!

Santa Clara Systems, Inc.
1860 Hartog Dr., San Jose, CA 95131
(408) 287-4640 / Telex: 176309 SCS SNJ

PCnet is a registered trademark of Orchid Technology
Pascal Bubble is a registered trademark of DATALEX

CIRCLE NO. 197 ON READER SERVICE CARD


Meet sweet IRMALINE.™
Your local operator for long distance mainframes.
Introducing IRMALINE, the interface who puts your personal computer or terminal on line with data bases in your IBM mainframe, with just a local call. No matter where your personal computer or terminal is. At home, for example, or in a branch office.
IRMALINE, the newest Decision Support Interface™ from IRMA's folks, is another industry first—a coaxial cable link between personal computers in remote locations and IBM 3270 controllers. With an old-fashioned modest price.
TM

She works with the big names.

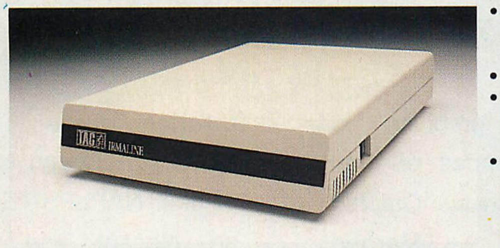
IRMALINE is a compact, self-contained protocol converter you can use with the IBM PC, the IBM PC XT, the Apple Lisa and the DEC Rainbow, to name just a few. She also works with IBM 3101 and DEC VT100 terminals.

Get on line with IRMALINE.

IRMALINE makes it possible—and very economically practical—for your personal computer or terminal to get on line with the mainframe from anywhere there's a phone. It's just a local call for the data you need.

IRMALINE,™ the newest Decision Support Interface™ from IRMA's folks.

- Coaxial cable protocol converter which works with most IBM 3270 controllers—3274, 3276, and Integral using Type "A" Terminal Adapters.
- User port provides RS-232C asynchronous data using DEC VT100 or IBM 3101 protocol.
- Total protocol compatibility with IBM 3278 and 3279 terminals.
- High speed TTL microprocessor handles the 3270 protocol independent of the RS-232C microprocessor.
- Buffered with full 3440 character screen page and extended attribute buffers.



- Emulates a 3278-2 using a standard asynchronous terminal or personal computer emulating a DEC VT100 or IBM 3101 terminal.
- RS-232C data rates from 110 to 9600 baud.
- Coaxial pass-through port allows an IBM 3270 terminal (or IRMA) to operate normally when RS-232C port is not in use.
- When used with TAC's IRMALETTE™ Asynchronous Interface, the IBM PC, IBM PC XT, and IBM PC compatible machines may run standard IRMA™ software for communication with IRMALINE.™

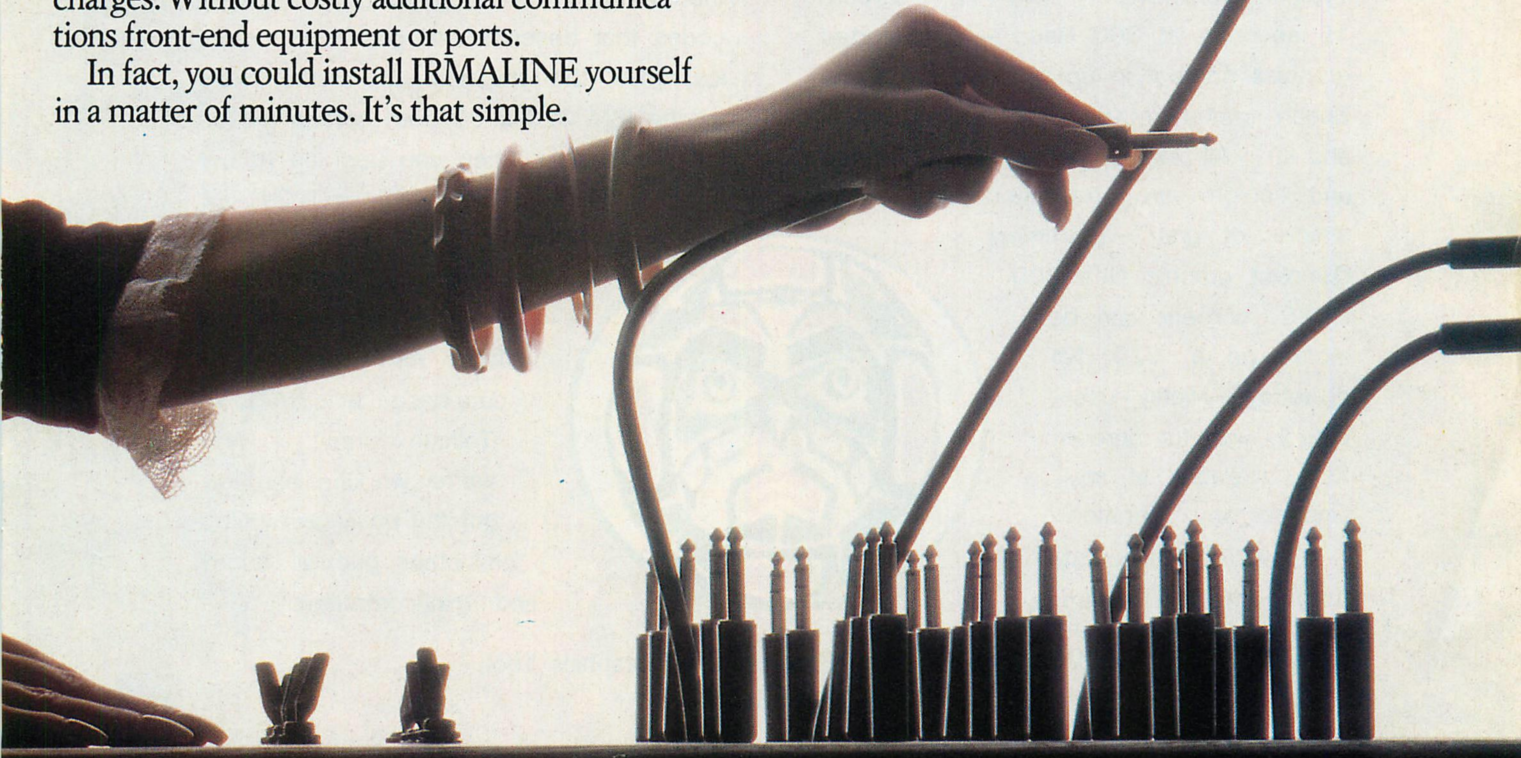
With just a local call, she lets you access mainframe data. If you're working with a personal computer, you can store that data and send reports back.

You do all this from wherever it's most convenient to work. Without expensive long-distance charges. Without costly additional communications front-end equipment or ports.

In fact, you could install IRMALINE yourself in a matter of minutes. It's that simple.

For more information about IRMALINE write to TAC, 120 W. Wieuca Rd., NE, Atlanta, GA 30042. Phone (404) 252-1045, Telex: 54-9600. Or you can call toll-free (800) 241-IRMA.

CIRCLE NO. 210 ON READER SERVICE CARD



Getting computers down to business.

TAC

TELIgraph

The Universal Graphics Tool

FEATURES: Sophisticated graphics in minutes, with 8 colors, 7 shades of gray and almost limitless textures. Graphic objects are macro-definable — Merged graphics and text: 7 character sizes are provided, including all international characters and special symbols. Characters can be colored as well as rotated and located on any pixel of the screen — Full communications facilities for graphic pages. Typical transmission time per screen: 15 seconds at 300 Baud — undistorted graphics printout in up to 5 sizes on any Epson printer (MX80/82/100, models II and III). All aspect ratios are corrected and colors are translated as shades of gray — Extremely compact graphic files: 300 to 500 screens can be stored on a 320 Kb diskette — Highly user-friendly, with full screen help available at any time, for any operation — Extensive & accurate documentation includes tutorial guide, and reference index. **PLUS:** TELIgraph is the full implementation of the NAPLPS-TELIDON World Standard, an elegant and powerful concept for graphics creation and transmission. This standard has already found thousands of professional applications: TV animated commercials, computer assisted education, design, flowcharting, games, etc.... TELIgraph does not generate a bit image of the screen, but a stream of machine codes that describe the generated graphics. Thus, the graphics you create today on the IBM 320 x 200 dot screen will be fully compatible when you upgrade to our 512 x 512 (or 1024 x 1024) dot systems. Plus, TELIgraph allows you to transmit and receive pictures to or from ANY machine or data-base that is NAPLPS-Telidon format — and we're working hard to get the package running on other popular micro and mini-computers.

When we say this is the Universal Graphics Tool,
we mean just that!

Keyboard-Input version: $398.95 (CANADA: $495.95)
Graphics Editor - Decoder - Communications - Slide Show
Ask your local dealer or order directly from Microtaure
MasterCard - Money Order - Certified Cheque

Intelligent professional graphics tablet version available soon.
Tablet resolution: .001 inch - Sizes: from 11" x 11" to 42" x 60"
Dual RS-232 ports, stylus & crosshair cursor included

microtaure

CIRCLE 328 ON READER SERVICE CARD

Minimum System requirements: IBM Personal Computer with BOTH color & monochrome screens - 256K Memory - One disk drive

MICROTAURE Inc.,
P.O. Box 6039, STATION "J"
OTTAWA (ONT.) K2A 1T1
(613) - 745-6661

TECH JOURNAL
VOL. 1, NO. 2

Directions

ON THE MATTER OF PERFORMANCE

What You See May Not Be What You Get

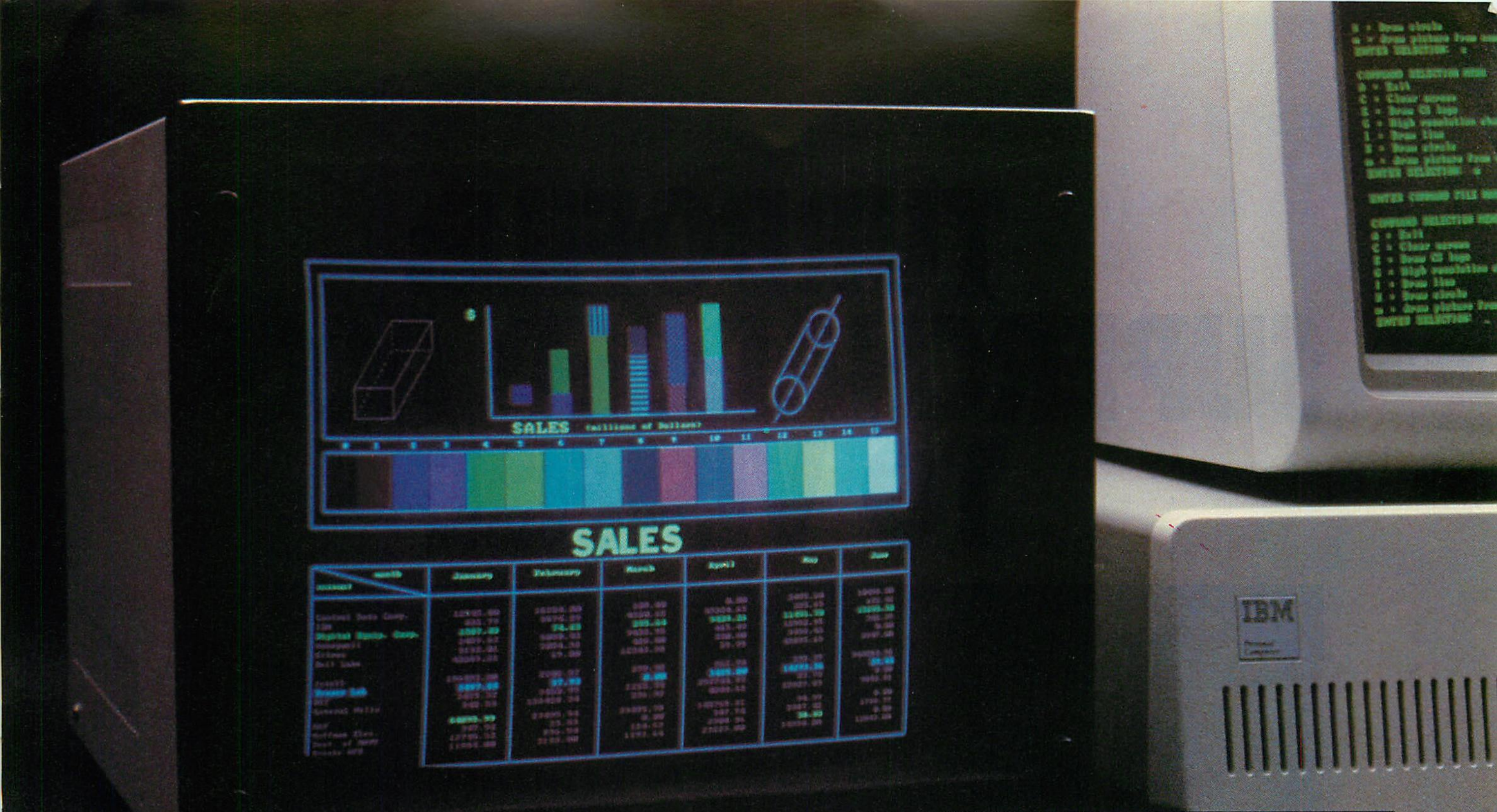
WILL FASTIE

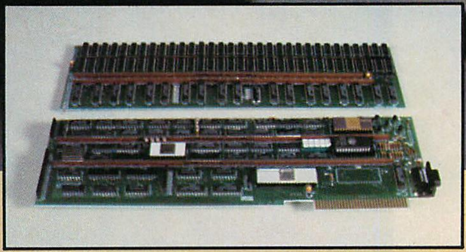
The PC is the source of an interesting contradiction in terms. At its introduction, critics (including yours truly) raved about the potential of the first mass-market small computer with a "16-bit" processor, and extolled the virtues of the PC architecture with its huge memory space. Almost immediately, those same critics (yes, again) complained about the performance of the software available on the PC. Particular targets were interpreted BASIC and VisiCalc; both products were in wide distribution and therefore easy to compare.

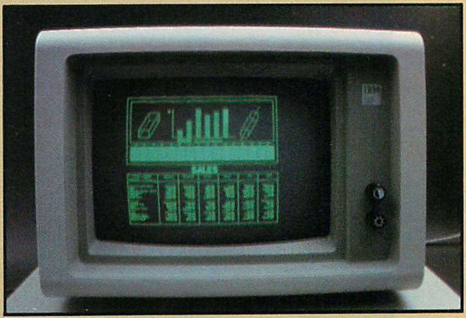
In fact, there is no contradiction between the *potential* of the hardware system and the *actual performance* of PC software. It is the software that must be examined to understand why such a popular machine has been so roundly cursed for its performance. There are a number of factors to consider.

THE NEED FOR GOOD PERFORMANCE

There are many applications for which outstanding performance is an absolute requirement. Examples are real-time, process-control, and data collection systems, which often need to make every machine cycle count; numerical or statistical applications, with their requirement for intense computation; graphics and animation; and many other CPU-intensive programs. It is probably safe to say, however, that IBM Personal Computers engaged in such activities account for no more than 5 percent of all PC's. Which is to say, of course, that 95 percent of the installed machines have very little need for very high performance. *(Cont. page 14)*







ARTIST

Two High Performance Graphic Controllers for the IBM Personal Computer.

ARTIST transforms the IBM-PC into a graphics work station that would sell for over $30,000. Tektronix 40XX emulation software allows the IBM-PC to interface with mainframe graphics software.

Output frequencies are adjustable for any monitor. ARTIST can drive the IBM monochrome display with 16 levels of intensity and 11 patterns of shading.

ARTIST has its own graphics library accessible from any programming language under PC-DOS or QUNIX. ARTIST also supports CP/M-GSX which provides communication to printers, plotters, and digitizers.

FEATURES

- **16 colors**
- **16:1 display zoom**
- **16:1 character zoom**
- **Pan, scroll, paging**
- **Light pen**
- **DMA**
- **Mixed text and graphics**
- **Selectable character sets**
- **Solid & dotted lines**
- **11 shading patterns**
- **RS343 output**
- **External genlock**
- **NEC 7220 processor**
- **Single expansion slot**

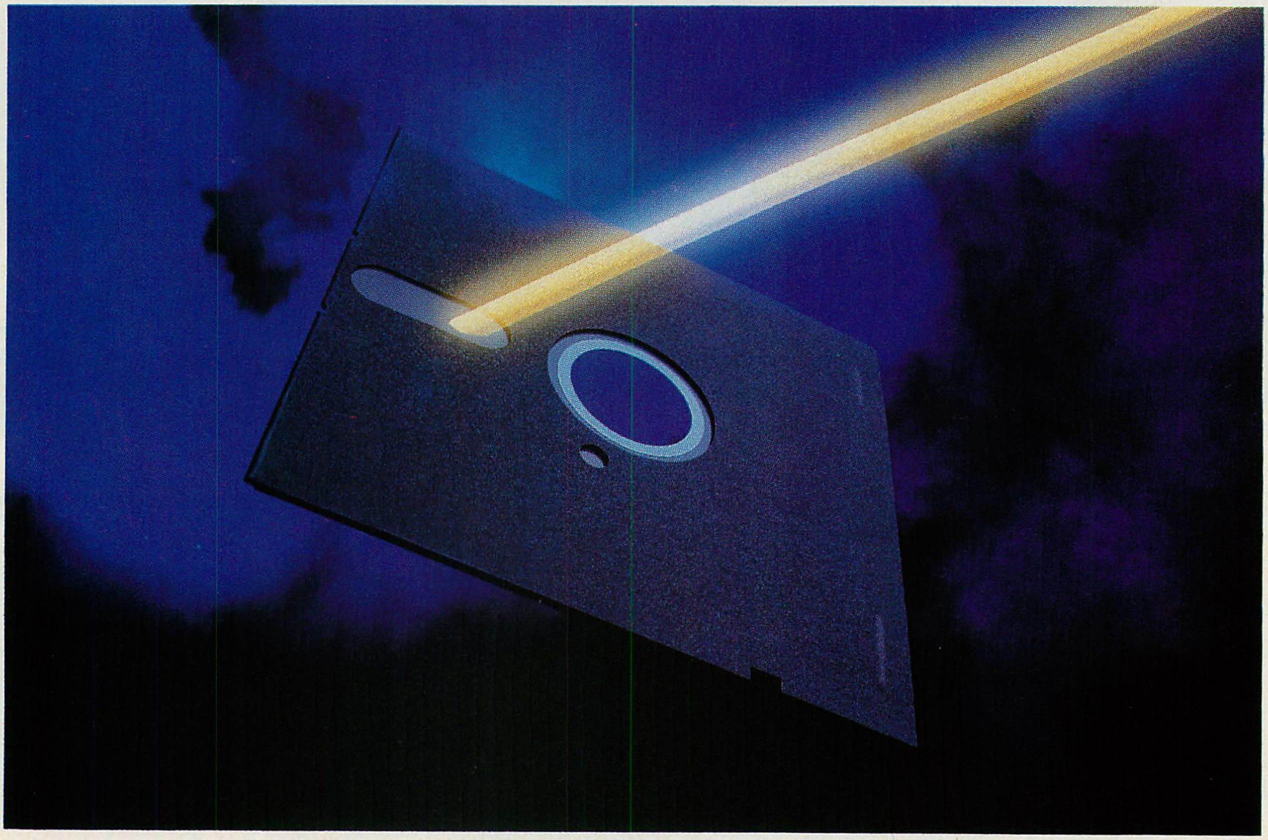
ARTIST 1 / $3195
1024 x 1024 Industrial graphics
170 x 96 Character display
512K Memory
16 - 40 MHz Bandwidth

ARTIST 2 / $1595
640 x 410 Business graphics
80 x 50 Character display
128K Memory
16 MHz Bandwidth

CS
CONTROL SYSTEMS 2855 Anthony Lane, Minneapolis, Minnesota 55418 (612) 789-2421

CIRCLE NO. 132 ON READER SERVICE CARD

TURBOFILE: SUPERCHARGED ACCESS TO VISIFILE™



You know and we know that VisiFile is probably the finest data base management system on the market today. We wrote and designed it to be.

But somebody asked us, "Can't you help us access our files and information in special ways . . . for our special needs?"

So we designed TurboFile. It offers you customized access, by letting you write BASIC programs that can read and write VisiFile files. For example, TurboFile allows you to access up to ten files simultaneously. Quickly and easily, by full key, partial key, or record number (that's *real* cross-referencing access). Using the BTree indexes already built and used by VisiFile.

Unlike other DBMS systems, TurboFile uses the same data files as VisiFile. What that means is that along with all of the powerful and friendly capabilities of VisiFile, you add TurboFile's unlimited programmability.

The best part is that TurboFile is so easy to use. As easy as saying GET, REPLACE, ADD, DELETE, or UNDELETE. TurboFile adds about 15 extra statements to BASIC for a host of special functions. At the same time, it removes the complications of BASIC programming.

This is one DBMS that is going to make you a lot more powerful. Because in the world of the microcomputer user, access is power. And TurboFile gives you access.

TurboFile is currently available for the IBM Personal Computer, and is priced at $195. For additional information call Creative Computer Applications, Inc. (603) 888-6648, or write PO Box 7074, Nashua, NH 03060. We're the ones who wrote VisiFile.

VISA and MASTERCARD accepted.

CIRCLE NO. 134 ON READER SERVICE CARD

TURBOFILE

Save your fingers. Let ProKey™ do the typing!

ProKey saves you time every time you use your computer.

It gives you a quick and easy way to eliminate repetitive typing, automate routine tasks, and smooth out rough spots in your progams. With the touch of a single key you can issue a whole series of commands to a program. This frees you to concentrate on the creative, human aspect of your work, while ProKey deals with computer details.

The result: a clean and simple productivity gain. And more time for the things you'd rather be doing.

The more you use your computer the more you need ProKey. Start saving time now for only $75.00. See your local dealer or order directly from RoseSoft, (206) 524-2350. Please add $2.00 for shipping ($3.00 by air).*

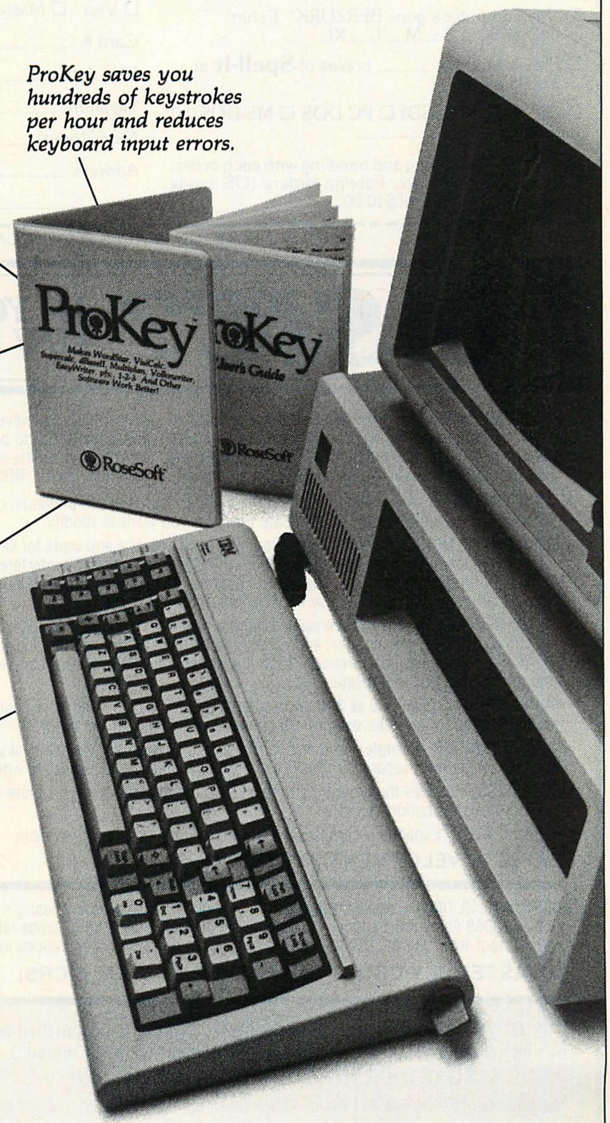
ProKey is your surrogate typist inside the computer. You tell it "When I type this, you type that." This is a single keystroke; that can be anything from one character to a complex sequence of commands.

ProKey saves you hundreds of keystrokes per hour and reduces keyboard input errors.

ProKey comes with Starter Sets™ for WordStar, Visicalc, dBASE II, and DOS. Address envelopes or print spreadsheets with a single keystroke.

ProKey is easy to learn and use. An online tutorial and well-written manual will have you defining your own "prokeys" in minutes.

ProKey saves your definitions in files you can edit and print. Have a special ProKey vocabulary for each of your programs.



RoseSoft™

4710 University Way N.E., Suite 601
Seattle, WA 98105 (206) 524-2350

ProKey is produced by RoseSoft—a company known for its professoinal, technical expertise; its excellent customer support; and its fast and efficient service.

**Check, money order, COD, Visa, or Mastercard accepted.*

CIRCLE NO. 108 ON READER SERVICE CARD

(Cont. from page 10)

This assertion can be quickly validated by considering the vast number of slower and smaller Apples and TRS-80s that have been so gainfully employed these past years in so many fields of endeavor, and how many are still so engaged. If performance were so critical, these machines would simply not have been suitable for the interactive demands placed upon them. Furthermore, some of the CPU-intensive applications can be completed by these machines just by letting them churn away by themselves, perhaps even while their owners sleep through the night.

So we must ask again: what makes performance an issue?

The Marketing Theory of Performance

It is very common to hear marketing and sales types lay all the performance issues at the foot of the micro-processor. The IBM PC gets a sound thrashing in such cases.

Most notable is the "68000 syndrome." To get this right, it is necessary that the salesperson thrust out the chin and raise the nose as high as possible; sniffing in a haughty way while talking is also most helpful. The victim is told of the obvious power advantage of the Motorola 68000 processor over the puny Intel 8088. And it is true: the 68000 *is* a more powerful processor. The IBM PC and its 8088 may be more than adequate for the task at hand, however, so a decision should be based on software and not on raw CPU power.

Less frequently, a potential buyer is subjected to the "Z80 effect." Here the salesperson takes on a mocking tone, humiliating the 8088 devotee not with claims, but demonstrations of software running faster, or just as well, on a Z80-based machine. And it is true: many tests have shown 4MHz Z80's running VisiCalc, BASIC, and other programs faster than the IBM PC. What is less obvious to the casual observer is the way in which the PC version of the software was implemented. Many programs are

written as they would have been on the Z80, and do not fully exploit the power of the 8088. The experience of some software developers, with programs specifically developed to take advantage of the 8088 architecture, indicates a potential performance advantage of two to one in favor of the 8088.

The 68000 folks are right, but it only makes a difference to the 5 percent who really need the steam. The Z80 folks are really right: software developers take heed.

The Theory of Apparent Performance

It is of utmost importance that software developers understand the matter of apparent or perceived performance if they wish to build software systems that meet the performance requirements of the computer user. It is equally important that the users understand the difference between what the program product can really do and what it *appears* to do.

VisiCalc, SuperCalc, and Multiplan are all full-function spreadsheet calculators with radically different performance characteristics. VisiCalc and SuperCalc display information on the screen at approximately the same speed, whipping columns and rows on and off the screen as the cursor is moved to and fro. When it comes to calculations, however, SuperCalc is about three times faster than VisiCalc. For larger models, this can be a very important consideration, while for smaller ones the difference is often not noticeable.

Multiplan, on the other hand, has a very sluggish display. Rows and columns kind of slide in, a datum at a time. Multiplan *appears* to be very slow indeed. At calculations, however, it is slightly faster than SuperCalc.

See the point? In a side-by-side comparison in a retail store, one that probably involves only small spreadsheets, SuperCalc and VisiCalc *appear* equivalent while Multiplan, the fastest of the bunch, *appears* painfully slow. In fact, SuperCalc has good display speed *and* good computational performance, a combination that neither VisiCalc nor Multiplan may claim. Figure 1 illustrates the point and helps to show that what the potential buyer sees may not tell the whole story.

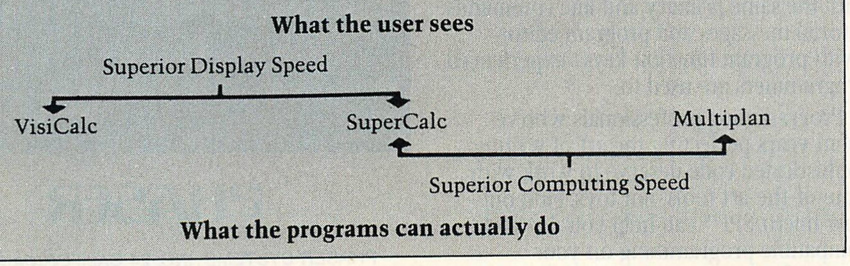
Getting What You Pay For

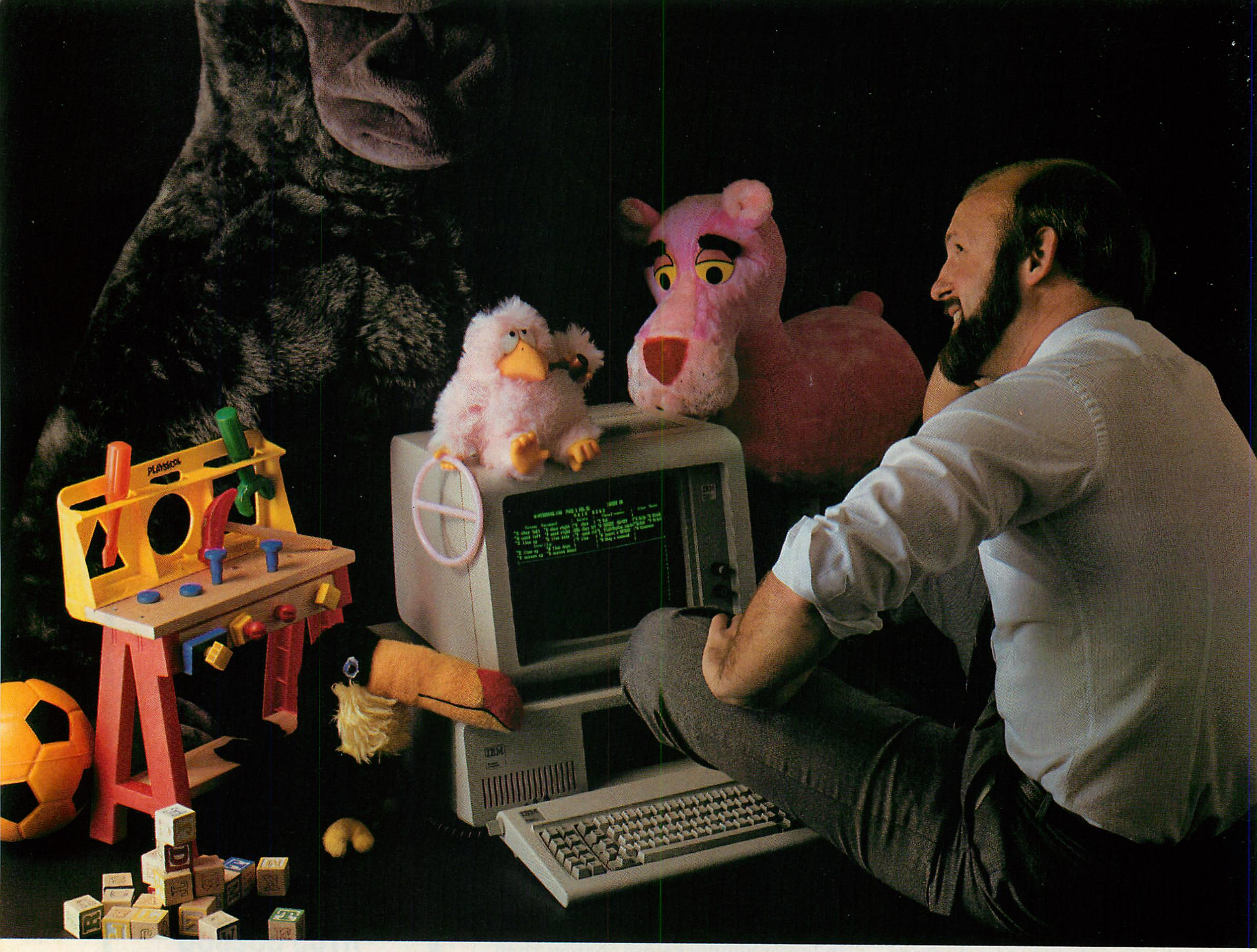
It's been said many times: Americans like the latest, best, biggest, fastest thingamajig. In cars and computers (and most everything else), this philosophy has a nasty way of backfiring.

It is really important to know for which functions the new gadget will be used. The twice-as-fast Epson FX-80 will dump a graphics image only a tiny bit faster than an MX-80 if the program doing the work takes 20 seconds between lines. The four-times-as-fast Hayes Smartmodem 1200 will not result in lower connect time charges than the Smartmodem 300 if The Source mainframes are so loaded that response time is long. The incredible 8087 math processor, 100 times faster than 8088/8086 software routines, is worthless if BASIC doesn't know how to use it and can even inflict a penalty if used badly. Each of these items commands a premium price: are they worth the extra money for the work they will do?

In 5 percent of the cases, the answer is yes. ■

Figure 1: Apparent Performance





This Programming professional deserves a lot more from his personal computer.

He's earned it. As a seasoned professional, he's learned to master some of the world's most advanced programming tools. Tools specially designed to meet the everyday demands of programming experts.

But as the owner of a personal computer, he's come to expect less. Less performance. Less sophistication. And less flexibility.

Why should programming a personal computer be any different?

Prior to the announcement of micro/SPF™ development software, experienced programmers felt programming a personal computer was a lot like playing with a toy. You couldn't take it seriously.

But today, there's micro/SPF,™ a solution to elementary program editing tools now offered with most micro-computers.

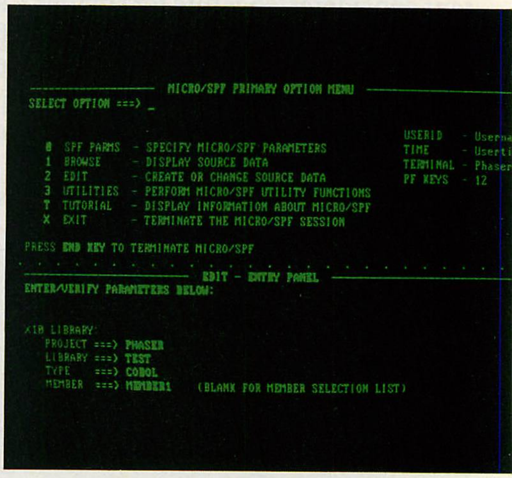
With micro/SPF™ you get the same procedures and commands experienced programmers are accustomed to using at work. By mimicking features found in standard SPF software, micro/SPF™ provides all the sophisticated utilities programming professionals expect.

Programming experts can take advantage of skills they've spent years perfecting.

Now, for the first time, mainframe software is available for personal computers. SPF screens are fully reproduced in logical sequence and each screen is formatted identical to those found in the SPF system.

In addition, micro/SPF™ comes equipped with the same primary and line commands, tutorial messages and program editor (with program function keys) experienced programmers are used to.

Programming professionals who've spent years perfecting the art of writing sophisticated code deserve to work with state-of-the-art tools, not toys. Find out how micro/SPF™ can help you do work-compatible programming on your personal computer today!



PHASER

PHASER SYSTEMS, INC. 24 CALIFORNIA STREET
SAN FRANCISCO, CA 94111

CIRCLE NO. 185 ON READER SERVICE CARD

CP/M® gives you a new world of PC power

...for a new low price.

Now, for just $60, you can have the world's most popular operating system for your IBM® PC. Only $60 for instant access to the largest collection of applications software in existence—literally hundreds of proven, professional software programs for every business and educational need.

New highs in productivity.

New enhanced CP/M-86® comes with print spooling, an exclusive feature that increases the work power of your IBM PC like no other operating system can. With print spooling you can print documents and run applications programs *at the same time*. And, for the ultimate in efficiency, all CP/M-86 programs are upward-compatible to Concurrent CP/M,™ the productivity breakthrough that lets your PC run as many as four different programs simultaneously!

New high-performance graphics.

Another unique advantage of enhanced CP/M-86 is GSX,™ the new graphics extension that allows CP/M to operate the latest graphics hardware devices. This lets you choose from a growing array of exciting, sophisticated graphics applications packages which run under the widely-accepted GSX standard.

Great high-level languages.

For programming under CP/M, Digital Research offers the most complete selection of professional-quality languages and utilities for the IBM PC: CBASIC® and CBASIC Compiler,™ Pascal/MT+,™ CIS™ and Level II COBOL,™ "C," PL/I, DR Logo,™ and more.

High time to buy!

Before now, you'd pay $250 for CP/M-86 and another $75 for GSX. That's what makes new enhanced CP/M-86 such a terrific value. At $60 it puts PC Power within reach of any budget! You'll find this enhanced CP/M-86 value in The CP/M Library,™ a selection of the finest in CP/M software for your IBM PC. Look for it today at your IBM PC dealer. Or contact Digital Research Inc., (408) 649-5500, 160 Central Avenue, Pacific Grove, CA 93950.

DIGITAL RESEARCH™

The best of everything in software.

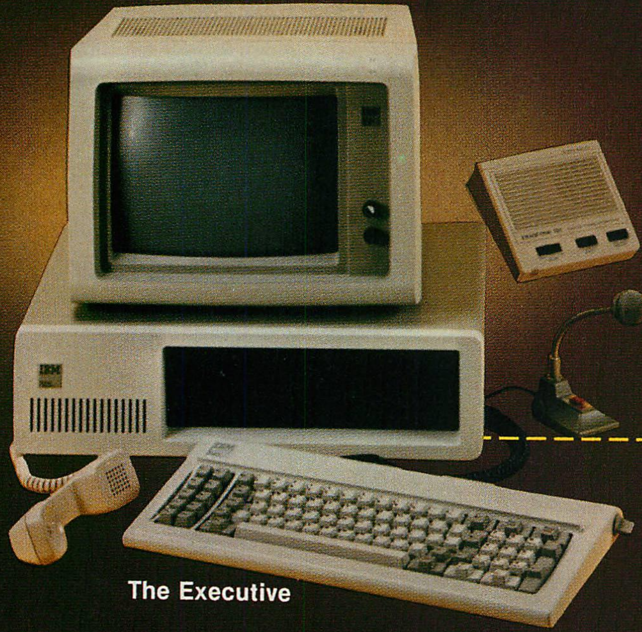
ELAN™* From TECMAR

The Most Powerful Local Area Network for the IBM PC

- Networking using Ethernet protocol
- Telephone support — autodial/autoanswer modem
- Voice recognition of commands and voice response
- Voice mail and messages
- Dictating system using Pearlcorder X-02 or XR from any telephone in the U.S.
- Complete software for written/voice mail, time management and audio/visual clock, calendar, and calculator functions
- Three ELAN Systems; the Secretary, the Manager and the Executive

The Manager

The MANAGER system adds a modem for telephone data communications and by adding a separate handset, will permit voice communications. The modem enables the MANAGER to receive unattended voice and data from any telephone in the U.S. Also, the MANAGER can accept commands through decoding the tones from the telephone keypad.



The Executive

The EXECUTIVE is the most complete implementation of ELAN, adding computer recognition of spoken commands. An executive might phone the PC to leave or retrieve messages or request specific information. The PC, in a spoken voice, can request a user's access code or prompt the user for a command. The executive can respond either by pushing buttons on the telephone, or by actually speaking back to the computer.

The SECRETARY is the basic ELAN system. It includes an Ethernet interface and all other ELAN features except modem and voice recognition.

ELAN is designed to meet your total communications needs, including computer-to-computer (i.e., networking), person-to-computer, and person-to-person communication requirements for data and voice. The industry-standard high speed Ethernet network protocol is employed in ELAN, permitting many IBM Personal Computers to be linked together by ordinary thin coaxial cable. In addition to his own computer, an ELAN system user can access the other devices attached to the cable — such as printers and large disks.

All versions of ELAN include an Ethernet interface with equipment to convert voice into data and back again. This enables the user to give and receive spoken messages from any location as well as store them for later use. In addition, with the Executive version, all ELAN software packages can be operated through verbal commands, through the telephone keypad or through the IBM PC keyboard. The computer can then respond verbally, either by telephone or 'in person'.

Whether you choose the EXECUTIVE, MANAGER or SECRETARY, an ELAN system will improve your productivity and expand the versatility of your IBM PC.

The Secretary

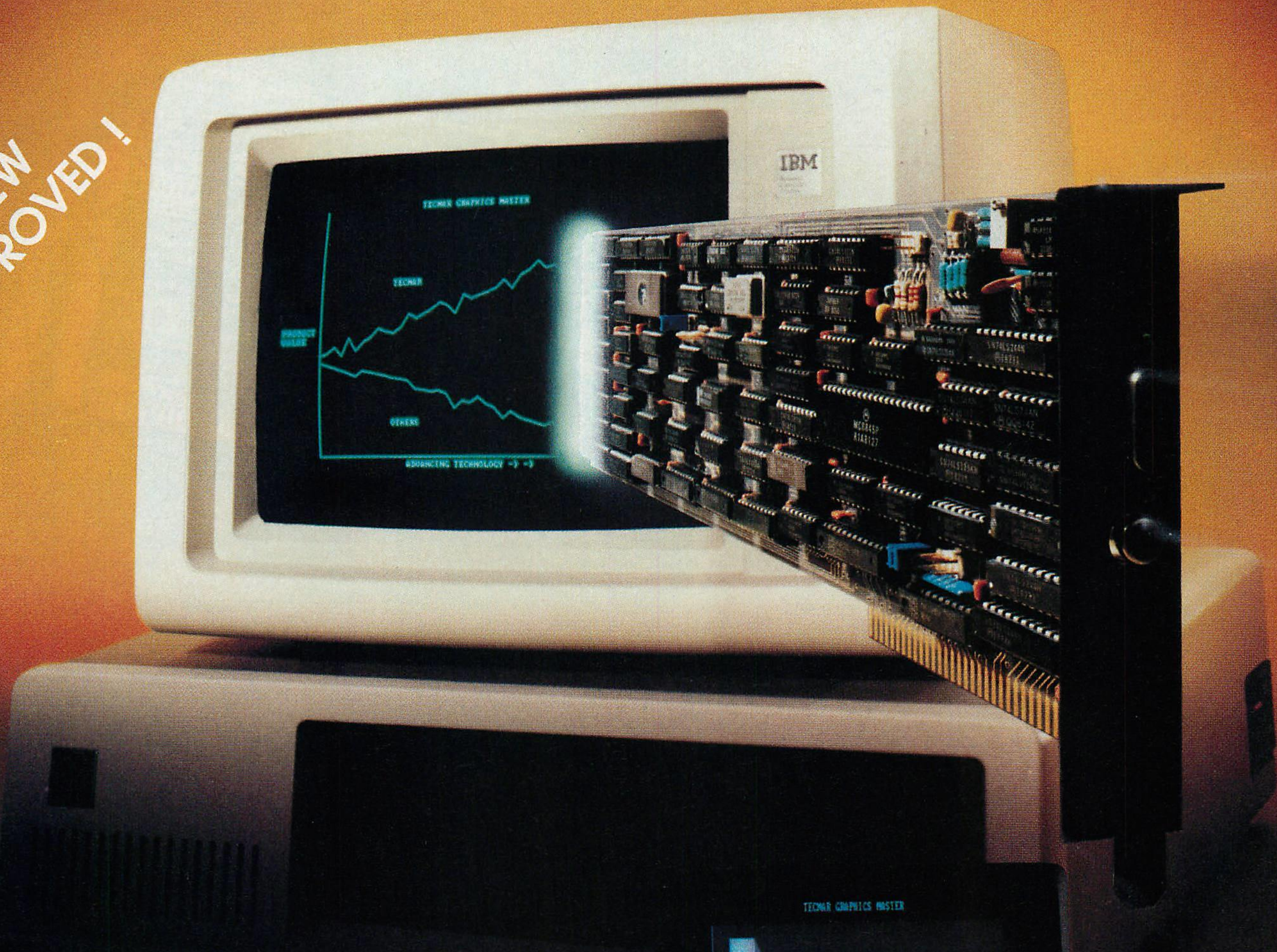
*ELAN (Extended Local Area Network) formerly ComNet

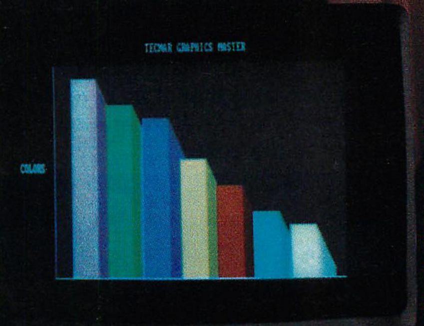
TECMAR 6225 Cochran Road, Cleveland, Ohio 44139, Phone (216) 349-0600, Telex 466692

CIRCLE NO. 200 ON READER SERVICE CARD

TECMAR'S GRAPHICS MASTER.™

It adds graphics to your IBM Monochrome Display, or 16 high-resolution colors to your color monitor!





Text *or* high-resolution graphics on the IBM Monochrome Display

Amazing new resolutions on standard color monitors

***Complete* emulation of IBM's display adapters**

On standard low-scan color monitors:
- 640 horizontal x 400 vertical in 16 colors
- 720 horizontal x 480 vertical graphics in 4 colors.

On the IBM Monochrome Display:
- 720 horizontal x 700 vertical
- 25/50 lines of 80 characters

Graphics Master can do much more. In all, over 30 different display modes can be used on standard monitors, and over 25 on the IBM Monochrome Display.

With certain monitors, you can achieve super-high resolutions like 512 horizontal x 420 vertical in up to 16 colors *without interlace.*

Graphics Master is equipped with a full 128K of on-board display buffer memory. In low-resolution modes, extra memory isn't wasted. You can divide it up into alternate display screens or use it for general data storage.

Graphics Master retains complete compatibility with the IBM Monochrome and Color/Graphics Adapters. Use it to accompany or replace one or both of the IBM cards.

Graphics Master will run with all IBM software for the Personal Computer, including BASIC's graphics commands.

With Tecmar's Video Cassette Recorder Controller, you can overlay Graphics Master images on videotaped material with an external mixer.

Call or write for further information. Our new address is:

TECMAR

6225 Cochran Road
Cleveland, Ohio 44139
Phone 216-349-0600/Telex 466692

CIRCLE NO. 290 ON READER SERVICE CARD

THE TECH JOURNAL NEWSLINE

Sol Libes

News, views and gossip on the IBM and IBM-like marketplace

IBM Upgrades 68000 System

A year ago last June IBM, via its Danbury, CT, Instruments Division, introduced its System 9000, a 68000-based 16-bit microcomputer for scientific laboratory applications. IBM has now upgraded this system to a real-time multi-user, multi-tasking system that emulates the Digital Equipment PDP-11 at 40 percent less cost. The system's base price is $5,695. Furthermore, IBM intends to sell the system via independent dealers. For the past year and a half the system has only been marketed as a laboratory instrumentation computer. It is also interesting to note that the DEC-like operating system software was not developed in house by IBM but instead came from Link-Data, Union, NJ, an IBM dealer for the System 9000.

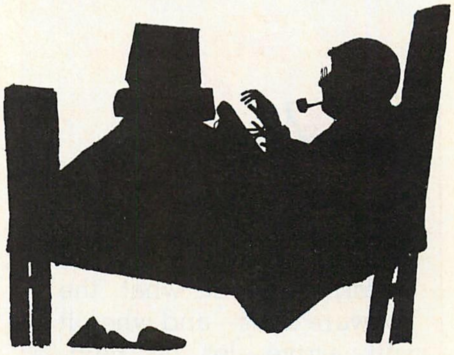
What all this means is not exactly clear. For one thing it means that IBM is significantly stepping up its competition with DEC in the minicomputer market that DEC has traditionally dominated. Secondly it means that the System 9000 will be in competition with the IBM-XT running the Xenix operating system, which is also a real-time multi-user, multi-tasking system. Xenix for the IBM-XT is currently being offered by Microsoft and not by IBM directly and hence IBM may not view this as competing with itself.

IBM Impacts Apple Sales

There are increasing reports that the IBM-PC is having an adverse effect on sales of the Apple IIe, particularly to small and large business users. Although the Apple IIe still dominates the home user market and still is very strong in the educational marketplace, its position in business usage has dropped significantly since the introduction of the IBM-PC and its look-alikes.

The first sign of this change in the marketplace came late last year when Apple and Computerland, the largest chain of computer retail stores, severed their long-standing relationship, in favor of carrying the IBM-PC and Compaq computers. Apple then entered into agreements with most of the Computerland stores to sell to them directly. Now, it has become apparent that many of the Computerland stores are dropping the Apple line as these stores change from what were previously consumer oriented retail outlets to business oriented outlets. Apple sales at one time comprised 80 percent of the volume of Computerland stores and currently is only 20 percent.

While Apple sales continue to grow in end-user and educational



markets, the sales to large and medium size corporations has decreased, and these companies are increasingly turning to the more powerful versions of the IBM-PC and XT machines. Increasingly, IBM has been exploiting its formidable position in installed mainframe computers to land large corporate sales for the PC/XT via its direct-sales force.

The Apple III, which Apple had hoped would compete with the PC and XT, although meeting with some measure of success has not done really well, and its sales curve has pretty much flattened out. The primary problem appears to be a lack of software support from independent software vendors. Although the Lisa is expected to strengthen Apple's hand in corporate accounts, the $10,000 price tag and the small number of dealers for it has created concerns.

The IBM reputation together with a very aggressive advertising campaign has resulted in IBM dominating the business marketplace. Further, many corporate customers expect a better connection between IBM mainframes and the PC in the future.

NEWSLINE

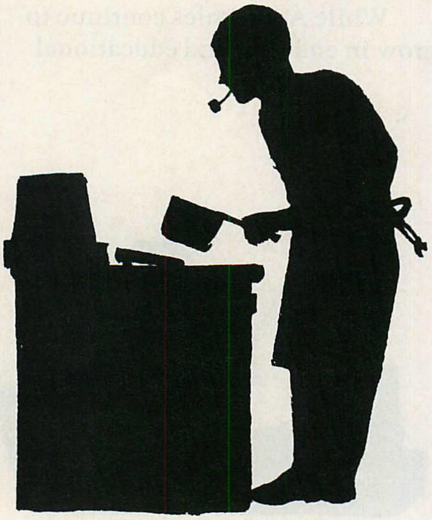
IBM Leads in Ad Expenditures Too!

IBM heads the list when it comes to spending money on advertising, which could be why they have the best image in the industry. Last year IBM reportedly spent $19.7 million to advertise the PC; Commodore was second spending close to $12 million, while Texas Instrument spent $8.8 million and Tandy spent $5. Poor Apple . . . they only spent $2.7.

Random Rumors

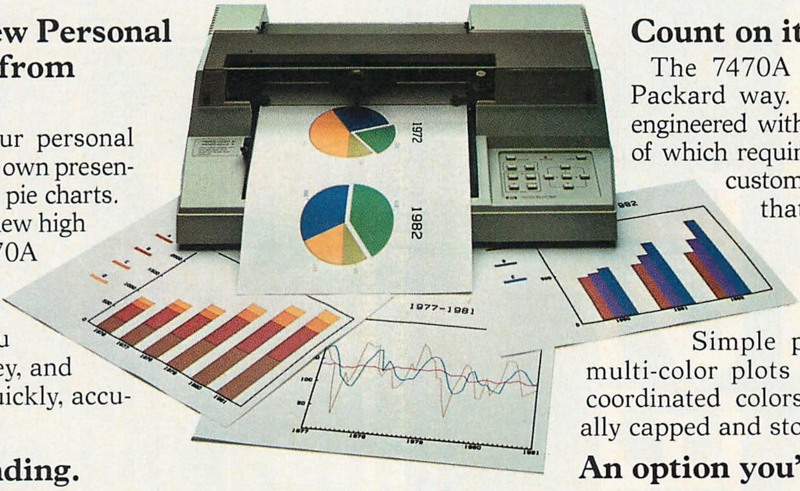
Watch for a new home-oriented computer to be introduced by IBM, possibly as soon as October. The unit is expected to be upward compatible with the PC and sell for between $700-800. It is expected to be a "transportable" machine, slim, lightweight, and includes a disk drive and 64K or 128K of RAM, *but no expandability.* Word is that they have already begun talking to mass merchandisers about the product. The problem is how to market the product without upsetting their current dealer organization and yet enter the larger home computer market . . . . Apple is expected to also introduce a similar machine at

about the same time . . . . Compaq is rumored readying two new versions of its machine. One will have a hard disk drive in place of one of the floppies. The other will have a 25 line by 80 character flat panel display (plasma type) in place of the CRT. Thus it should be considerably smaller and lighter than the current unit — and more expensive. Compaq is also expected to go public sometime soon . . . . Microsoft is expected to release a version of Xenix, its version of Unix, for the XT very soon. Xenix is a real-time, multi-user, multi-tasking operating system that is currently popular on large minicomputers . . . . However, IBM is expected to release a new operating system for the PC and XT that was developed in-house and is upward compatible with its larger mainframes . . . . and look for Tandy to release shortly an IBM-PC compatible machine and undercut the market in price. ■



How two pens can become your best presentation tools.

NEW LOW PRICE ONLY $1,095



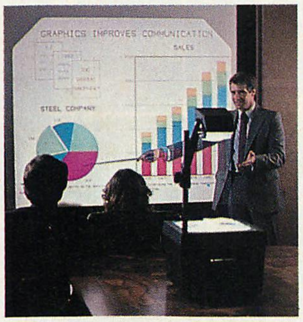
Introducing the New Personal Computer Plotter from Hewlett-Packard.

Now you can use your personal computer to generate your own presentation charts, graphs, and pie charts. How? Simply add on the new high quality, low cost HP 7470A Personal Computer Plotter.

The 7470A helps you save time and save money, and lets you communicate quickly, accurately and *effectively.*

Quicker understanding.

Data, when visualized graphically, becomes information fast. Charts and bar graphs can make any presentation clearer and more readily understood. But asking your staff to produce the graphics manually for your next presentation doesn't ensure accuracy or artistic talent. And going to outside graphics suppliers can be costly. Combined with your personal computer, the new HP 7470A plotter does the communicating for you. Quickly. Logically. And with off-the-shelf software available from most HP dealers.



Fast and pretty.

The 7470A gives you high plotting speed with excellent line quality...faster than any competitive small plotter. On top of all that, it comes in an attractive design package that looks nice on your desk. And it does it for only $1,095, (U.S.A. domestic suggested retail price.)

Count on it.

The 7470A is built the Hewlett-Packard way. To last. Designed and engineered with only a few parts, none of which require adjustment. And with customized integrated circuits that ensure reliability.

Pen pals.

The HP 7470A has two single-pen stables. Simple pen changes give you multi-color plots in your choice of ten coordinated colors. Pens are automatically capped and stored.

An option you'll want, too.

For only $95, you can also get a 17057 Overhead Transparency Kit that turns your plots into transparencies for overhead projectors. For "I need it tomorrow at 9:00 A.M.!" meetings, it's a necessity.

Start plotting your next presentation today. Clip and mail the coupon below. Now.

Mail the coupon below and we'll send you—absolutely free—a sample plot, a more detailed brochure, and a sample overhead transparency.

Then...stop in at your nearest Hewlett-Packard Dealer. See the HP 7470A in action. Once you see it demonstrated you'll find a hundred ways to make your own applause-winning presentations.

When performance must be measured by results

HEWLETT PACKARD

Seeing is believing. Send me a sample plot, an overhead transparency, and more detailed information.

Name ______________________ Title ______________________

Company ______________________

Address ______________________

City, State & Zip ______________________

Phone Number (   ) ______________________

My computer is ______________________

11203PT9

Send to: Hewlett-Packard, 16399 W. Bernardo Drive, San Diego, CA 92127—Attn: Marketing Communications

CIRCLE NO. 104 ON READER SERVICE CARD

EASIBOARD™

THE EASIEST™

1985 technology in 1983. It took longer than expected for our Research and Development Department to meet its challenge: create the most advanced multifunction board for the IBM PC – ahead of present day technology, and more advanced than the original multifunction boards introduced with last year's technology. And above all, make it simple and make it easy. The result: Easiboard by Easitech.™

Nine distinct functions in just one IBM PC or XT slot. After surveying business executives and other personal computer users, we discovered that nine computer functions were most desired. We merged them all together, and made Easiboard as simple as plugging in your tv at home.

Memory, Spooler, Printer and Modem Connections, Three Electronic Disks, Chronograph and More. Easiboard gives your IBM PC or XT more memory (up to 256K), three high-speed solid state electronic disk emulators, a printer connection, a communications port for modems or additional printers, a buffer-spooler so you can compute and print at the same time (part of the Easiboard memory becomes a memory for your printer), a battery-backed real time chronograph (an extremely accurate clock-calendar), and the ability to switch between two printers when both are connected to one PC. All on just one simple plug-in board.

IBM quality plus Easitech Excellence exclusively at IBM PC dealers worldwide. Easiboard is only available at authorized IBM PC dealers. You'll find Easiboard at the same place you find the IBM PC – quality retail stores you can depend on to carry only the best microcomputers and enhancements.

Easy to find the best price, too. Try $325 for Easitech Excellence and a two year guarantee. No need to search through dozens of magazines for low prices. Our lowest prices are at IBM PC dealers. Not anywhere else. And we back our product with a two year guarantee. If your IBM PC dealer does not carry Easitech products, ask him to call us at (404) 923-3186.

SPECIFICATIONS
EASIBOARD WITH EASIMASTER™ SOFTWARE

- Socketed for 64K to 256K of Random Access Memory.
- Three high-speed solid state electronic disk emulators, each many times faster than normal disk drives.
- Centronics-compatible parallel port for printers or other parallel devices.
- Serial RS232 async communications port for modems, printers or other serial devices.
- Extremely accurate chronograph (computerized clock/calendar in real time with Nicad battery back-up).
- Buffer/spooler which allows use of the computer while printing is underway, eliminating the usual waiting.
- Easiswap™, a feature which allows switching from one printer to another, as when both a dot matrix and a letter quality printer are connected to the same IBM PC. Or use it for switching communications devices.
- Ultra high-speed Motorola RAM chips, the gold ones.
- 100% IBM PC and XT compatible.

Easiboard with 9 functions. $325 (socketed for 256K). **$395** (64K installed, socketed for 256K). **$595** (256K installed).

IBM PC DEALERS

ALASKA • **Computerland**, Anchorage, Fairbanks, **Microage Computer Store**, Anchorage, • **ALABAMA** • **Computerland**, Anniston, Birmingham, Mobile, **Anderson Computers, Inc.**, Florence, Huntsville, **Olensky Brothers Computers**, Mobile, • **ARKANSAS** • **Computerland**, Fayetteville, Fort Smith, Jonesboro, Little Rock, • **ARIZONA** • **Computerland**, Mesa, Phoenix, Tucson, **Data Place Computer Stores, Inc.**, Phoenix, Tempe, **Photo & Sound Company**, Phoenix, • **CALIFORNIA** • **Computerland**, Anaheim, Bakersfield, Belmont, Burlingame, Capitola, Carlsbad, Concord, Dublin, El Cerrito, Fairfield, Fremont, Fresno, Glendale, Hayward, Laguna Hills, Lawndale, Long Beach, Los Altos, Los Angeles, Monterey, Newhall, Newport Beach, Novato, Oakland, Pasadena, Redding, Sacramento, San Bernardino, San Diego, San Francisco, San Gabriel, San Jose, San Leandro, San Rafael, Santa Barbara, Santa Clara, Santa Maria, Santa Rosa, Stockton, Thousand Oaks, Torrance, Tustin, Ventura, Visalia, Walnut Creek, West Covina, West Los Angeles, Woodland Hills, **Computer City**, Brea, Orange, San Diego, Santa Ana, **Gateway Computer Center**, Cerritos, Huntington Beach, **Photo & Sound Company**, Costa Mesa, El Monte, Fresno, Sacramento, San Diego, San Francisco, San Jose, **Exectec**, Los Gatos, Napa, **Computer Ware, Inc.**, Modesto, **Macys Computer Solutions**, Palo Alto, San Francisco, San Mateo, **Microage Computer Store**, Pleasant Hill, Santa Ana, **Businessland, Inc.**, San Jose, **Rac Computers, Inc.**, San Jose, **Computer Plaza**, Santa Barbara, • **COLORADO** • **Computerland**, Arvada, Boulder, Colorado Springs, Denver, Englewood, Fort Collins, Grand Junction, Greeley, Lakewood, **Compushop Inc.**, Aurora, • **CONNECTICUT** • **Computerland**, Danbury, East Hartford, Fairfield, Hamden, Stamford, W. Hartford, **Microage Computer Store**, Danbury, Greenwich, **Computerease**, East Hartford, Milford, **Entre' Computer Center**, Glastonbury, Orange, **Computerworks**, Westport, • **D.C.** • **Computerland**, Washington, **The Math Box, Inc.**, Washington, • **DELAWARE** • **Computerland**, Newark, Wilmington, • **FLORIDA** • **Computerland**, Altamonte Springs, Boca Raton, Clearwater, Coral Gables, Fort Myers, Ft. Lauderdale, Jacksonville, Lakeland, North Miami Beach, Orlando, Sarasota, Seminole, Tallahassee, Tampa, West Palm Beach, **Entre' Computer Center**, Altamonte Springs, **ICS**, Coral Gables, **The Computer Factory, Inc.**, Miami, • **GEORGIA** • **Computerland**, Athens, Atlanta, **Compushop**, • **Computerland**, Athens, Atlanta, **Compushop**, Atlanta, Chamblee, **Lenox-Davison's**, Atlanta, **BAS, Inc.**, Waynesboro, • **HAWAII** • **Computerland**, Aiea, Honolulu, Kaneohe, **Photo & Sound**, Honolulu, • **IOWA** • **Computerland**, Cedar Falls, Cedar Rapids, Davenport, Des Moines, Marion, **Omni Computer**, Des Moines, **Team Electronics**, Waterloo, • **IDAHO** • **Computerland**, Boise, **Golden W. Idaho Falls, Inc.**, Idaho Falls, **Silver Creek Irrigation**, Picabo, • **ILLINOIS** • **Computerland**, Arlington Heights, Bloomington, Champaign, Chicago, Downers Grove, Elmhurst, Evanston, Joliet, Mundelein, Naperville, Niles, Northbrook, Oak Lawn, Oak Park, Peoria, Rockford, Schaumburg, St. Charles, **Compushop, Inc.**, Buffalo Grove, Chicago, Midlothian, Morton Grove, Oak Park, Rolling Meadows, **Computermart**, Carbondale, **Main Street Computer Company**, Decatur, **Computer Corner**, Edwardsville, **Illinois Valley Irrigation, Inc.**, Havana, **Microage Computer Store**, Orland Park, **Computers Etc., Inc.**, Posen, **Computerland/Springfield Corp.**, Springfield, • **INDIANA** • **Computerland**, Anderson, Evansville, Fort Wayne, Indianapolis, Merrillville, Mishawaka, West Lafayette, **General Microcomputer, Inc.**, Elkhart, Michigan City, South Bend, **Chester Inc.**, Francesville, **Microage Computer Store**, Indianapolis, **The Computer Room**, South Bend, **Hoosier Electronics Computer Center**, Terre Haute, • **KANSAS** • **Computerland**, Hutchinson, Lawrence, Leawood, Topeka, Wichita, **Amerisource**, Prairie Village, **Microage Computer Store**, Salina, • **KENTUCKY** • **Computerland**, Bowling Green, Lexington, Louisville, **CBM**, Lexington, **CBM Computer Center**, Louisville, • **LOUISIANA** • **Computerland**, Baton Rouge, Lafayette, Metairie, New Orleans, **Computer Concepts**, Lake Charles, • **MASSACHUSETTS** • **Computerland**, Boston, Reading, Springfield, Wellesley, **Entre' Computer Center**, Dedham, North Darmouth, **Neeco**, Needham, **Microsource/Financial, Inc.**, Watertown, **The Computer Place, Inc.**, Worcester, • **MARYLAND** • **Computerland**, Annapolis, Lutherville, **Entre' Computer Center**, Baltimore, Timonium, **The Logical Choice**, Baltimore, Pikesville, **HLA, Inc.**, Beltsville, **Frederick Computer Products, Inc.**, Frederick, **The Math Box, Inc.**, Rockville, Wheaton, • **MAINE** • **Computerland**,

Envision a single high-speed communications board with integrated easy-to-use software that emulates any terminal.

It's not just your imagination. It's Linkup.™

At ITI, we know your data communications must be fast, accurate and inexpensive. Our engineers faced the same communications problems you face. That's why they designed Linkup™ for the IBM PC.

With Linkup, you have a high-speed, programmable communications processor integrated with user-oriented software that provides cost effective access to an international network of host computers, terminals and other personal computers. You'll be able to transfer text, graphics, facsimile images, telemetry and program files as easily as dialing a phone number.

Communication modes support high and low speed access to computing and data base services, electronic mail, order entry, credit verification, tele-conferencing and direct marketing.

Linkup is the only communications hardware you'll ever need.

A single plug-in module with the modular software emulates TTY, ASCII block mode, IBM 3101, IBM 2780/3780, DEC VT 52/100 and IBM 3270 and 5250. Terminal emulation modes are changed by a single keystroke without changing diskettes and switching modem cables. And most importantly, the same familiar user command set appears in all modes.

Linkup grows to meet your changing needs.

Future terminal protocol requirements will integrate into our programmable coprocessor. And, Linkup's high-speed data throughput allows you to take full advantage of increases in common carrier transmission line speeds.

Linkup is cost-effective.

Linkup's single communications board does more than several others combined, leaving your computer's precious board slots open for further applications. And its powerful MC 6809 processor chip handles all communications tasks, freeing the PC to service the user interface.

Faster, more accurate transmission means lower costs.

Linkup's integrated design architecture reduces connect charges by transmitting and receiving data at speeds up to 19.2 Kilobaud asynchronous and up to 56 Kilobits per second synchronous without the loss of data throughput normally associated with such speeds. Transfer files at high speed to the PC's disk, then print off-line. And, each line can be checked for error as it's received automatically, retransmitting only those lines in error, instead of entire pages. That saves connect time and character transmission charges.

Our innovative fill-in-the-blanks technique overcomes the tedium of question-answer interrogation and menu approaches.

The user simply enters phone numbers, passwords and job file names on a consistent fill-in-the blanks form. Communication options and commands are displayed on the screen form and selected with a single keystroke.

Linkup works around the clock so you don't have to.

Linkup automatically executes tasks. It dials, signs-on, runs a job unsupervised and maintains a log of its activities. It uploads, downloads and does printouts while you're away. And when you return, your data is ready and waiting, so you can put your time to its best use.

If you want to communicate, call ITI toll-free 1-800-431-3460 for additional information.

iti

Information Technologies Inc.
7850 East Evans Road
Scottsdale, Arizona 85260
(602) 998-1033

IBM PC is a trademark of International Business Machines Corporation.

Linkup Specifications

Speed: 56 Kbps Synchronous, 19.2 Kbaud Asynchronous
Terminal Emulation Modes: TTY, ASCII Block Transfer, DEC VT 52/100, IBM 3101, IBM 2780/3780, and future options include 3270, 5250, Tektronix 40XX.
File Types: ASCII, EBCDIC and Binary.
Automatic Modes: Dialup, Log-on, Job Execution, Answer, Alert, Unattended Job Scheduling.
Communications Interface: Dual channel RS232/423 compatible.
Modems: 103, 212, 201, 208 Bell compatible, Hayes, Vadic, Rixon smart modems, or user defined.
Bus Interface: IBM PC bus compatible.
PC Configuration: PC DOS, 64 Kbytes RAM, 160 KB Floppy, monochrome display minimum. Also supports IBM compatible hard disks, color displays and printers.
Options: Printz™ Print Spooler, Postmaster™ Electronic Mail, Audit™ Billing Summary
Base Price: $795 each, including TTY and Asynchronous block software.

CIRCLE NO. 154 ON READER SERVICE CARD

Letters

IBM Right/Wrong

My issue of *PC TECH JOURNAL* came in the mail, and I am very pleased. It immediately takes its place in the upper ranks of the IBM-PC-oriented computer magazines. If you can grow from here, you have a real winner on your hands.

About the "What IBM Did Right/Wrong" article: hey, the IBM keyboard is not all that bad. I have seen a lot of ink shed over that backslash key, and I can understand that people who are used to the Selectric keyboard would have a problem with it. But those of us who are used to the 3278 keyboard have no problem there.

What drove me wild the first few weeks that I had my PC were the function keys. They are on the opposite side of the keyboard! The 3278 has 12 keys on the right against the PC's 10 on the left. If you want function keys on the top of the keyboard, you can use the same trick that IBM uses to get 24 "PFkeys" on the 3278—use ALT and the numeric keys at the top of the keyboard. It requires no more programming tricks than getting the usual Function keys does.

I started out defending the PC keyboard, and now I've got a list of faults almost as long as the one in your article. Have I contradicted myself? No, I think not. Most of the faults are relatively minor as I now happily swap between a 3278 by day and my PC by night. I can live with the differences. What I hope I have made clear is that no one keyboard can be faultless to all people. Maybe IBM should come out with some optional keyboards for special users. A Word Processing Keyboard that exactly apes the Selectric layout would silence the most vocal detractors.

Carrington Dixon

PC Tech Journal Right/Wrong

I looked forward with great interest to the first issue of *PC Tech Journal.* My appraisal in one sentence: the editorial content of this issue paid for my entire one-year subscription, but the layout of the magazine stinks.

Hoffmann's article ("A Technical Review of the IBM Color Graphics Adapter,") contained information that can be found nowhere else. This is a real gem. The article on the XB precompiler ("The Anatomy and Construction of XB") had some good points, and the stereoscopic images article ("Computer-generated Stereoscopic Images") was worth reading. Bill Hunt's article on C Compilers ("How to Choose a C Compiler") was the type of article I expect from a technical journal and will be an excellent reference when I need it. The Legal Brief ("The IBM Software License") was also worthwhile. Thus, I feel that the content of the magazine was worth the (outrageous) $3.95 cover price.

I can live with the three-column format and the uneven column lengths; I can understand why you have chosen not to right-justify the columns. I can even come to terms with the excessive amount of advertising. But it is wrong to break up technical articles. It is a sin to place listings at the end of the magazine. It is an error to give incorrect page numbers in the index, and it is distressing to have an incorrect page reference at the top of a listing. These are not elements of style to be forgiven and lived with. These are just plain goof-ups that must be corrected before the Tech Journal becomes something more than a vehicle for advertisements—a tool for serious programmers, systems designers, and forward-looking software and hardware companies.

These are strong words, meant to have an effect; I would hate to see the Tech Journal die. The name "Technical Journal" means that you can avoid all the junk that is the domain of the other IBM-PC magazines. I really hate interviews with executives and "home banking is the thing of the future" articles. Technical also means that you must pay extra special attention to the accuracy and format of your articles.

Dan Rollins
Glendale, CA

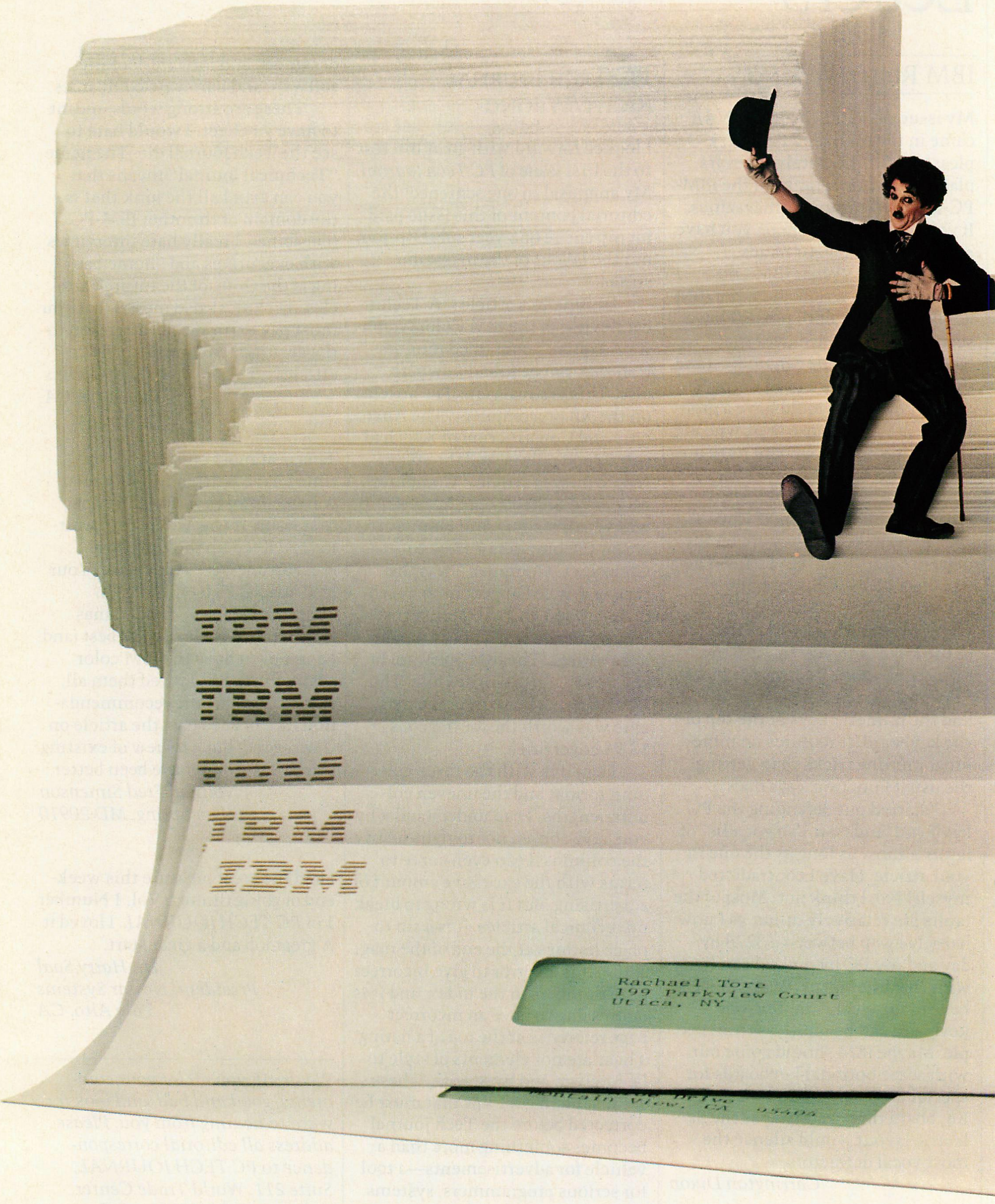
This is at most only the second time I have written to a journal. I waited almost a full year for the first issue. It was well worth waiting for. Keep up the good work: great articles, great coverage. Your lead article, "The IBM Color/Graphics Adapter," by Thomas Hoffmann was clearly the best (and I don't even have the IBM color adapter yet). But I loved them all, except that specific recommendations are needed e.g., the article on C was good, but a review of existing compilers would have been better.

Norman Fred Simenson
Silver Spring, MD 20910

I had the great pleasure this weekend of going through Vol. 1 Number 1 of *PC TECH JOURNAL*. I loved it. A great job and a great start.

Dr. Harry Saal
President, Nestar Systems
Palo Alto, CA

We thank you all for your comments, good and bad, and look forward to hearing from you. Please address all editorial correspondence to PC TECH JOURNAL, Suite 211, World Trade Center, Baltimore, MD 21202.—WF


IBM
IBM
IBM
IBM
IBM
Rachael Tore
199 Parkview Court
Utica, NY

8087 Performance Considerations

The 8087 can improve performance, but only if applied with care and understanding

NEIL SARNAK AND DR. ERIC JAFFE

Intel designed the 8088 as a low cost, general purpose microprocessor chip to serve a wide variety of computing needs. Among the features omitted from the 8088 because of cost and extra complexity in both design and fabrication were the ability to perform complex mathematical calculations and I/O in hardware. The 8087 Numeric Data Processor (NDP) and the 8089 I/O Data Processor were designed as optional co-processors that could be installed along with an 8088 to extend the original chip's instruction set and functional ability. While in hardware the 8088 can only perform simple arithmetic operations on 5 digit (2 byte) integers, the 8087 in hardware can rapidly perform a wide variety of arithmetic and transcendental operations on integer and real numbers as large as 18 digits (10 bytes). Though IBM personal computers are delivered with only an 8088, they contain an adjacent empty socket intended for an 8087 NDP. Once installed, the 8087 in the presence of appropriate ancillary software is almost as transparent to the high-level language programmer, though execution speed for programs that are computationally bound is markedly shortened. In other words, an 8088/8087 pair functions as though it were a super single processor.

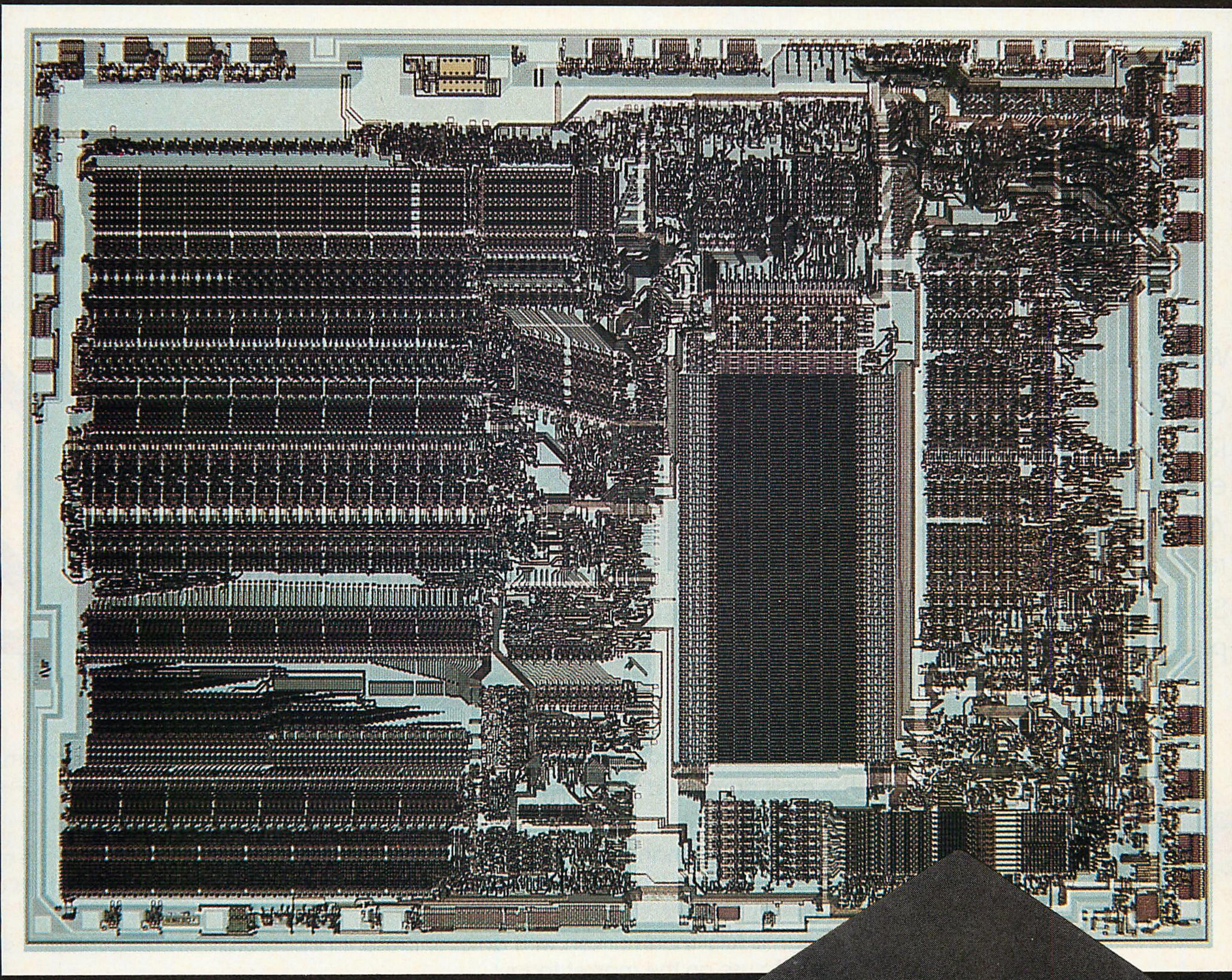
An 8088/8087 pair functions as though it were a super single processor.

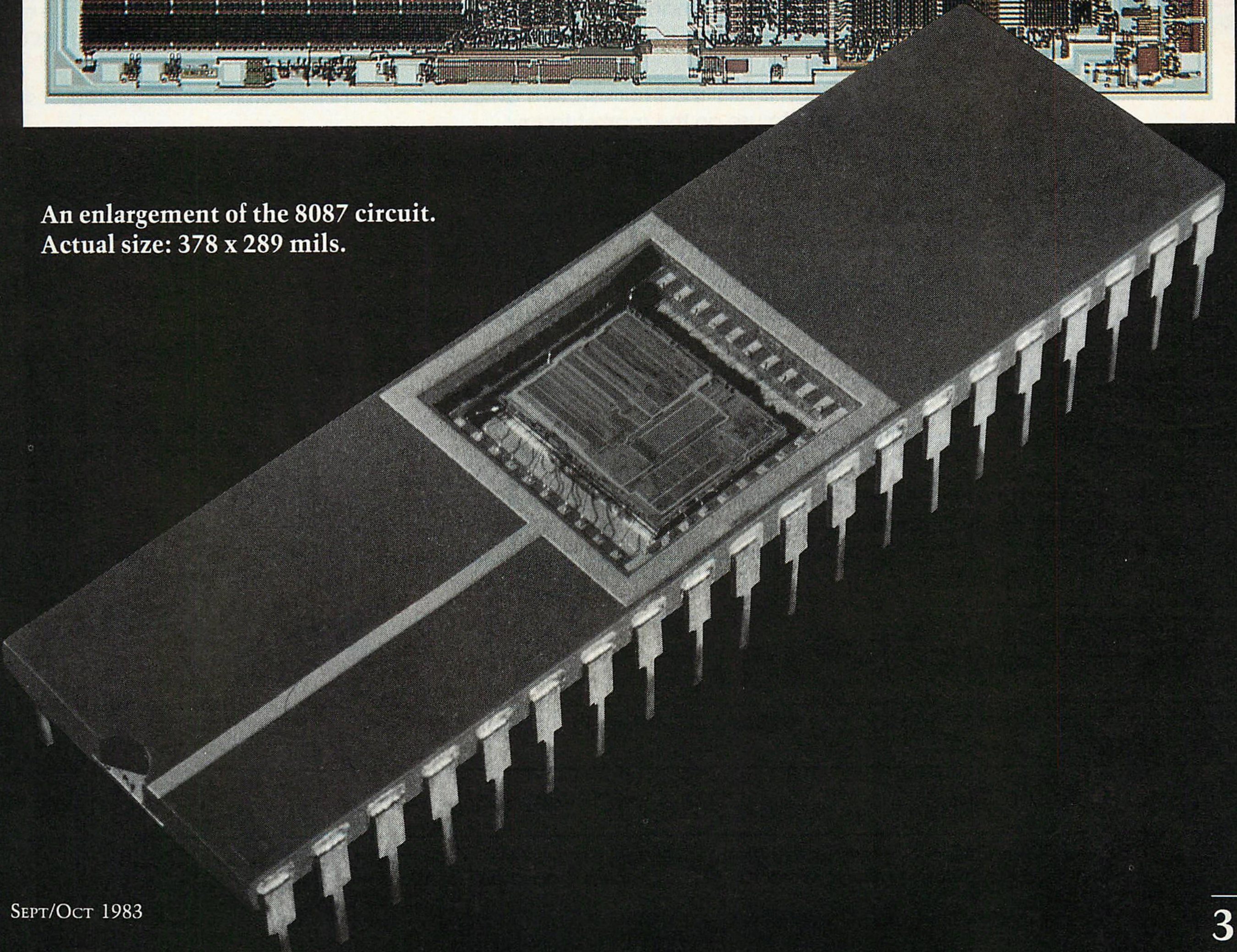
In order to understand how an 8087 NDP can so dramatically increase the capabilities of the 8088, one must know which arithmetic operations, comparisons, and data types the 8088 can and cannot support in hardware. Those functions not directly supported in hardware or whose data types are larger than the registers in the CPU being used must be implemented, at least in part, by software using those functions available in hardware. In the case of arithmetic operations, these software routines are often called arithmetic emulators, since they emulate arithmetic operations in software. Typically, arithmetic emulation routines use numerous 8088 hardware instructions to emulate one 8087 instruction and therefore execute more slowly, especially for complicated operations such as exponentiation.

Arithmetic operations may be performed on essentially two major categories of data: binary numbers

Neil Sarnak is a Ph.D. candidate in computer science at New York University's Courant Institute. His field of study is the analysis and design of graph algorithms. His interests include game design and development and applications using the IBM PC.

Eric Jaffe, M.D., is a professor at the Cornell University Medical College. Currently the president of the New York IBM Personal Computer User's Group, he has worked with computers for the past 20 years.





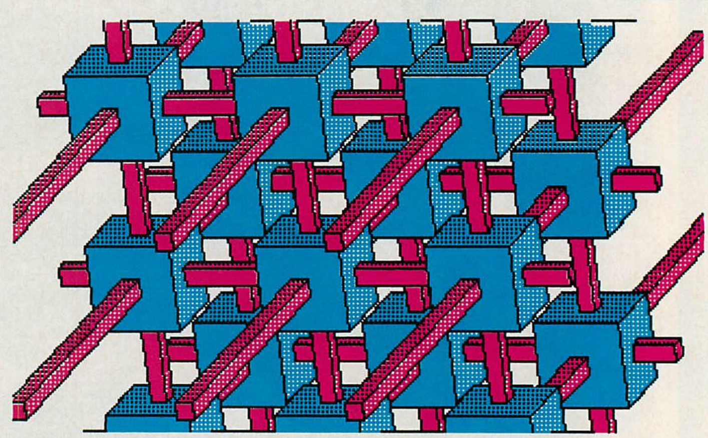
An enlargement of the 8087 circuit.
Actual size: 378 x 289 mils.

COURTESY INTEL CORPORATION

INTRODUCING A WHOLE NEW DIMENSION FOR MICROCOMPUTERS

3D!esign

THE POWER OF TRUE CAD IS NOW AVAILABLE FOR YOUR IBM PC



With 3Design you can:

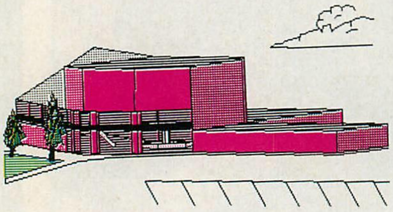
! Create you own 3-D structures

! Rotate, move and scale them independently or in groups in any coordinate system

! Achieve visual realism with fast hidden line removal

! Save, retrieve and duplicate data on disk

! Produce presentation quality output



Features

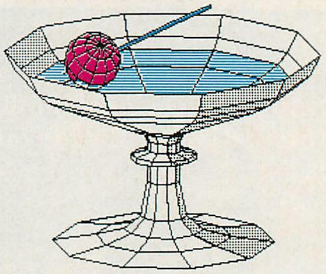
! The most advanced hidden line removal anywhere (an extended NASA algorithm)

! The most advanced graphics relational database available

! Clipping, zooming, panning

! Objects can be changed at any time with the friendliest graphics EDITOR you have ever used (functions such as rubber line, point matching, gridding and many others are provided)

! 3Design is fully menu driven with an ever-present HELP feature, making it the easiest system to use on the market today



! ARTISTE presentation package allows color mixing, shading and text with a standard display

! Three-dimensional data can even be sent over the phone lines, if you have a communications interface and software

! Structures can be defined by three-dimensional functions (i.e. Z=X•Y)

! Optional hard-copy output capability with most printers (full color with IDS Prism*) with the I-C Magic* chip

! You are even protected against power failure. 3Design creates constant back-ups

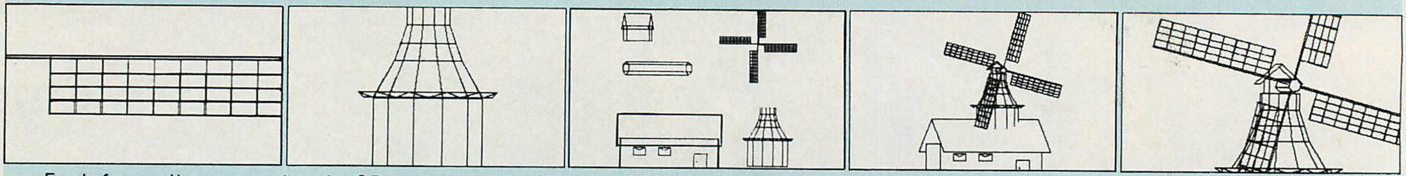
! Optional WordStar* interface will let you intermix figures and text

! **From a circuit board to a city block, 3Design can lay it all out for you; quickly, easily, professionally**

$249
128KB version
($299 with an I-C Magic)

$80
64 KB version
($129 with an I-C Magic)

Prices include all the software described above plus utility programs, documentation and sample structures. Minimum system configuration —64KB, color card, 1-320KB disk drive.



For information on orders for 3Design, call toll-free 1-800-392-9210 (in Washington call (206) 525-7820)

Or write: 3Design, 4710 University Way N.E., Suite 1512, P.O. Box C-56789, Seattle, WA 98105

!

*Prism is a registered trademark of Integral Data Systems. I-C Magic is a registered trademark of Microcomputer Business Industries Corp. WordStar is a registered trademark of MicroPro International Corp.

and binary coded decimal (BCD) numbers. The category of binary numbers can be subdivided into signed and unsigned numbers, while BCD numbers may be split into packed and unpacked numbers. All these data types are understood by the processor to represent integer numbers. The main difference between binary and BCD data types is in their representation of integers. The binary data type assumes that each bit in a number represents a corresponding power of two. In other words, binary numbers are simply the binary representation of the actual number. In contrast, the BCD number types assume that every group of four bits represents one decimal digit. Thus, two decimal digits may be stored in one byte, depending if the number is packed or not.

In hardware, an 8088 chip can perform addition, subtraction, multiplication, and division on the four types of numbers described above. These operations may be performed as a single operation only on numbers whose representation is no longer than two bytes (sixteen bits). This results in unsigned binary numbers ranging from 0 to 65535, and signed binary numbers ranging from - 32768 to 32767. Similarly, BCD numbers can range from 0 to 9999. For those applications that require only numbers that fall in the above range, the 8087 will provide only a minor improvement in speed. For example an 8088 can multiply two, two-byte numbers in 32 microseconds whereas an 8087 can perform the same task in 27 microseconds.

However, the above ranges of numbers are obviously inadequate for those applications requiring basic arithmetic operations on integers larger than 65535 or real numbers or calculation of various transcendental functions.

To remedy this problem, many computer languages, including most of those in use on IBM PCs, contain emulators that perform arithmetic and transcendental operations on real numbers. The number of bits used to represent a real number determines the accuracy that can be achieved. The more bits used, the better the accuracy; but for any given microprocessor, the longer the execution time whenever the number of bits used is larger than the CPU register size. The most common emulators (e.g. those provided in BASIC) can emulate operations on 32 bit and 64 bit real numbers. Thirty-two bit numbers are accurate to about seven decimal places. This means two numbers of equal magnitude (size), but differing in the eighth or more place will appear to the 8088 to be the same number i.e. 1.23456789 and 1.23456781 would be considered equal. Sixty-four bit reals give an accuracy of about 15 decimal places. It should be clear, given the 8088's limited hardware implementation of arithmetic operations (16 bit registers), that emulation of arithmetic operations on 32 and 64 bit reals will be substantially slower than the performance of simple arithmetic operations on 16 bit integers.

In summary, the 8086/8088 when used for numerical calculations suffers from two deficits: only a limited repertoire of numerical operations have been implemented in hardware, and those only work directly on numbers that are two bytes or less in length.

8087 NDP Operations

The 8087 was designed to overcome both the above problems. An 8087 contains eight 80-bit registers and has the ability in hardware to perform the four basic arithmetic operations as well as calculate square roots, tangents and arctangents, and logs and exponents to the base 2 on numbers up to 18 digits in length. Certain constants such as zero, one, π, log 2 to the bases $_e$ and 10, and the logs of $_e$ and 10 to the base 2 are part of the 8087's instruction set and can be pushed onto the 8087's stack in less than 5 microseconds. Associated functions such as other trigonometric operations and exponentiation to the bases $_e$ and 10 are supported by software routines that use the basic hardware functions. All operations performable in hardware can be executed in less than 140 microseconds and most operations take considerably less than that (addition - 18 microseconds, division - 42 microseconds). In addition, because of the

An 8088 can multiply two, two-byte numbers in 32 microseconds whereas an 8087 can perform the same task in 27 microseconds.

register length and internal design of the chip, it takes the 8087 roughly the same amount of time to multiply two double precision reals (36 microseconds) as it does to multiply two two-byte integers (32 microseconds). Since an 8087 takes 100 microseconds to perform an exponentiation, in contrast to the approximately 25 to 100 times longer to do the same operation with 8088 software, the utility of an 8087 for complicated and lengthy numeric calculations rapidly becomes clear. The 8087 can also compare all its data types.

In addition to substantially shortening the time needed for numeric computations, the 8087 numeric data processor adds one totally new numeric type, extends the size of three other types, and contains hardware instructions to support calculations on the new types. With the inclusion of an 8087, the following new data types are supported:

short integer - This is a 32 bit integer (the standard integer size on an IBM 370 mainframe), which allows nine digit accuracy and an integer range of -2×10^9 to 2×10^9.

long integer - This is a 64 bit integer. It provides 18 digit accuracy and an integer range of -9×10^{18} to 9×10^{18}.

packed decimal - This is the same as 8088 packed decimal but with 18 digit accuracy.

short real - This is a 32 bit real, with 7 digit accuracy and a range of 8.43×10^{-37} to 3.37×10^{38}.

long real - This is a 64 bit real with 16 digit accuracy and a range of 4.19×10^{-307} to 1.67×10^{308}. (In contrast, an IBM 370 mainframe allows about 10^{78}).

temporary real - This is an 80 bit real with 19 digit accuracy and a range of 3.4×10^{-4932} to 1.2×10^{4932}.

Until packaged software is re-written with the 8087 in mind, there may be no improvement in running time.

Thus, by using an 8087, several additional numeric data types become available and the resulting instruction set compares more favorably to that present in large computers. In both theory and practice, all these data types and their corresponding arithmetic operations can be emulated by 8088 software, but at the expense of increasing execution times by a factor of 10 to 100. Other operations such as square roots and sines would require another ten-fold increase in execution time if performed by software.

Practical Problems

You might therefore think "Enter the 8087 NDP, exit the waiting" and that if you install the 8087 chip, the running times of your programs can be improved by a factor of at least 100. Unfortunately, this is not true for several reasons.

In the worst (and most trivial) case, no improvement at all is seen. This usually occurs because the software (e.g. spread sheet program) is still performing arithmetic operations via emulation routines rather than using the 8087. Until packaged software is re-written with the 8087 in mind, this will probably be the usual state of affairs.

If the 8087 instruction set is partially supported, variable amounts of improvement will be obtained depending on which instructions are implemented and/or used. Most of the software packages to be discussed support certain 8087 data types and their corresponding arithmetic instructions. These usually include real numbers (either 32 bit or 64 bit) but not long integers, temporary reals, or packed decimals. Certain other 8087 operations on real numbers are also supported (e.g. square root, log, exp, and cosine).

8087 support is usually not provided by the original vendor of the compiler. Thus, the company providing 8087 support must work within the frame-work set up by the original compiler writer. The IBM DOS Pascal Compiler produced by Microsoft is a good example of the limitations of this type of arrangement. Since the code generated by the IBM DOS Pascal compiler only provides 32 bits of storage per real number, 64 bit real number support is impossible without re-writing parts of the compiler. This is the reason that so few of the 8087 numeric data types are supported.

Many compilers without 8087 support emulate real arithmetic by using code that jumps to the location of the appropriate emulation routine. To provide 8087 facilities, the jump destination is merely changed to a location where 8087 instructions are located. This method will obviously improve running times, but not as significantly as would be achieved by generating 'in-line' 8087 instructions which do not require jumping to another location.

The overhead involved in moving data between memory and the 8087 significantly slows execution speed. To add two 80-bit temporary real numbers, each would need to be loaded into a register. Since the IBM PC data bus is only 8 bits in size, this would require 20 transfers (2×10 bytes) along the bus to load the numbers and ten more to store the sum.

The need for pre- and post-processing of arguments of functions also slows expected execution speed. Computation of a sine is a good example. The only trigonometric functions available in the 8087 are partial tangents and partial arctangents and the 8087 will only accept an argument for these functions of $0 < \text{theta} < \pi/4$ and $0 < z < 1$ respectively. Thus, in order to compute a sine, the argument must be properly scaled to fit the aforementioned requirements, a partial tangent calculated, and then using trigonometric identities, a sine calculated. Obviously, this requires a fair amount of computation and takes a great deal more time than calculating just one function.

Lastly, arithmetic instructions do not stand alone in programs but rather are parts of complex loops that are often executed many times in the course of a calculation. These loops usually involve a large number of instructions for either control of the loop or movement of data compared to the number of instructions actually involved in doing arithmetic computations. The program used to generate the timings shown in Figure 1 is a good example. In Network Consulting (NCI) Pascal using an 8087 and the program converted to native code, one passage through the FOR loop which does addition of two double precision reals takes about 120 microseconds (performs 8380 adds/sec). Inspection of the assembly language listing of the program reveals that the actual addition by the 8087 takes about 10 microseconds and the control structure of the loop and transfer of the data to and from memory on the data bus takes 100 microseconds. Under such circumstances, use of a super chip that could perform the same addition operation in 1 nanosecond would shorten execution time of the loop by only about 10%. However, execution times could be decreased by increasing CPU clock speed, in-

creasing data bus width, decreasing number of assembler statements required or by optimizing compiler code.

Software

There are many language implementations that can support their standard data types using 8087 instructions including Ada, APL, BASIC, C, COBOL, FORTH, FORTRAN, LISP, Pascal, and PL/M. We have tested the add-on packages from Hauppauge Computer Works, Seattle Computer, and Microware.

Add-on packages from Hauppauge Computer Works, Seattle Computer, and Microware were tested.

These packages run under IBM DOS and work only on compiled programs in the appropriate language. The Network Consulting Inc. (NCI) 8087 support package is included with their version of the UCSD p-system and works with either single or double precision reals and with either interpreted or native code programs. The P-system compiles all source code into p-code (pseudo code) which is then interpreted at runtime. P-code can in turn be translated into machine (native) code.

Microware offers 8087 support packages for BASIC, Pascal, FORTRAN and assembler; we benchmarked the Basic and Pascal packages. The FORTRAN package was not tested because the FORTRAN compiler (as distributed by IBM) contains many bugs. The Hauppauge package provides Pascal and assembler support, while the Seattle Computer package provides BASIC and assembler support. The NCI P-system implementation supports any program run under the P-system. While these are mainly Pascal and assembler, BASIC and FORTRAN are also supported.

Installation

All the suppliers of 8087 software also supply the actual 8087 processor. All but one of the suppliers (Hauppauge) ship the 8087 as is. The 8087 as supplied by Hauppauge is installed in a carrier. This carrier is a good idea, because it helps protect the delicate pins of the 8087 during its installation into the empty socket of the IBM PC. The 8087 is installed by easing it into the socket, while taking care not to bend any pins. Unfortunately, once the 8087 is in the socket, it is difficult to determine if any pins were bent in the process. One installation manual suggests the removal of the 8087, a visual check for bent pins, and reinstallation; doing it twice might be asking for trouble. The carrier supplied with the Hauppauge package seems to prevent bent pin problems.

Installation is completed by flipping one switch on the PC system board (switch number 1, position 2). The installation instructions of all the packages are clearly explained. The Hauppauge and Microware packages also have software tests that validate the installation.

Usage

All DOS based support is invoked in a similar way. Compilation is done as usual, with only some small changes in the source code. At the link stage, the 8087 support package is linked into the code, producing executable code that uses the 8087 instruction set. The standard IBM DOS Pascal compiler and IBM DOS BASIC compiler used with the "/0" option create large overheads in terms of code size (approximately 30K bytes and 20K respectively). This means that a small Pascal program, say 1K bytes, will compile to an executable code file of about 31K bytes. The 30K bytes overhead does not include any 8087 support. The 8087 routines add approximately another 10K bytes.

The Hauppauge Pascal package supports only single precision reals and allows computation by an 8087 of the standard four arithmetic functions, square roots, the main transcendental functions (sine, cosine, tangent, log and exp), and a series of advanced transcendental functions. All the real number routines are supplied both in source and object code form. The source is assembly code and is useful because it demonstrates how the 8087 is used.

The Microware packages tested include BASIC and Pascal. The Pascal compiler will normally only support single precision real numbers, while the BASIC compiler supports both precisions. In order for the Pascal compiler to generate IEEE format real constants, the Pascal compiler's PAS1 routine must be patched. This modification is performed by Microware, so that one must send a diskette with the PAS1 file to Microware which they patch and return. To use Microware's BASIC compiler 8087 support you must link in the BAS87 code. Both single and double precision real numbers are supported.

The Seattle Computer package provides both BASIC and assembler support. Both precisions of real numbers are supported. The compiler must be invoked with the "/0" option. This option creates a 'stand-alone' code file that does not require the runtime support of the compiler. Unfortunately, this restricts the use of chaining with the COMMON command.

The Network Consulting Inc. (NCI) version of the P-system supports both precisions of reals and the basic and transcendental operations on real numbers. Unlike the other packages, NCI's 8087 support works with either interpreted or native code versions of a program. Since the UCSD P-system Pascal package as distributed by NCI has four separate interpreters and matching sets of "realop" subroutines which calculate transcendental functions (single or double precision reals, 8087 or software math) and two different native code generators (8087 or software math), you

See us at:

PC '83 · SC SOFTCON

BOSTON
OCT 8-10
BOOTH A441

NEW ORLEANS
FEB 21-23
BOOTH 1151

(Read all about it.)

Low cost TEXTRA™ makes the competition see red...

Ever since the days of the vacuum tube, the computer industry has travelled at the speed of light, and the result is clear—*more power at less cost.*

So now, when Textra makes word processing *faster & easier to use* at a fraction of the cost, it's not surprising—*just another sign of the modern times.*

See it for yourself

Textra has created a software revolution because we let you protect your money decisions until you've tested the product. Our TEST DISK contains an 'on-line tutorial' and a copy of Textra that you can use for testing the features. It costs $3, and we'll even refund $2 if you send us the disk back. See our ordering instructions below.

Take a closer look

When we say faster, we mean instantly. Paging, searching, reformatting—they all work instantly, speeding you through your work. Textra is easier to use because it's an original design for the IBM Personal Computer. It has labeled function keys instead of awkward "control codes", full use of the cursor moving keys, and a list of features that just won't quit.

Features

- full screen editing
- modeless insert and delete (overstrike and undelete also available)
- powerful search and replace
- copy and move blocks, merge from file.
- tabs, margins, word wrap, auto-indent, paragraphs, line shifting and centering
- extensive reformatting abilities.†
- full printing capabilities including print-while-you-edit, page numbering, headers, footers, and much more.†
- numerous other features which make your time at the keyboard more productive
- word processing or program editing

†Features not available in Textra Jr.

Watch our documentation

Relax. Sit back and watch it. Dim your lights a little and let it roll. Our 'on-line tutorial' is a series of short films which automatically roll by on the screen, showing you how to use Textra. In less than an hour, you'll be using Textra efficiently. Of course, a thorough reference manual is included if you have any questions.

CAUTION: Textra is addictive. If you already have another word processor, you might be into antique collecting.

$95

☐ Send me a copy of Textra for $95
☐ Send me a copy of Textra Jr. for $39.95
Michigan residents add 4% sales tax.
Please add $4.00 for postage and handling.
☐ Send me the Test Disk for $3.

Name__________
Address__________
City_____ State_____ Zip_____
☐ Visa ☐ Mastercard
#__________ Exp. Date_____

CIRCLE NO. 227 ON READER SERVICE CARD

textra jr.

An Outstanding Text Editor.
An Outstanding Price. 39.95

Ideal for:
program editing • simple word processing • dBase II programming • works with Visicalc, Supercalc, Lotus 1-2-3 files.

P.S. You can upgrade to Textra at anytime for the difference in price.

***A² Software**

*Ann Arbor Software, 407 N. Main Street
P.O. Box 7345, Ann Arbor, MI 48107

Order: 313-769-9088
(COD, Visa/MC, Check/Money Order.)

Textra also works with the PC XT and leading compatibles. Textra requires PC DOS and 64K. (128K recommended, required for DOS 2.0) and a monochrome or color monitor.

Textra, and the A² Software logo are trademarks of Ann Arbor Software Associates. Wordstar is a registered trademark of Micropro, Inc. Powertext is a registered trademark of Beaman-Porter, Inc. Easywriter II is a trademark of IUS.

FIGURE 1

Benchmark of Pascal and BASIC Programs, running IBM, Hauppauge 8087, Microwave 8087, Seatle 8087, and NCI software. All numbers are calculations/second, except Hausdorf, which are in min:sec

PROGRAM	ADD	SUB	MULT	DIV	LOG	EXP	SQRT	SIN	HAUSDORF
IBM BASIC INTERP2	294	296	278	221	105	118	108	73	17:30
IBM BASIC COMP2	946	920	865	800	206	227	538	246	2:35
IBM BASIC INTERP4	250	240	192	47	102	112	105	71	26:02
IBM BASIC COMP4	890	882	566	498	81	84	228	74	4:15
MICRO BASIC 287	5000	5000	5100	5000	2500	2200	5005	1200	1:20
MICRO BASIC 487	4990	5000	4995	5000	2500	1600	4990	1200	1:45
SEATLE BASIC 287	1485	1463	1463	1415	1250	968	1485	915	1:28
SEATLE BASIC 487	1224	1205	1200	1190	1053	872	1316	843	1:52
IBM PASCAL COMP2	4000	3261	2308	750	69	53	30	53	3:02
MICRO PASCAL 287	5660	5357	5455	5000	1007	1000	5357	490	:35
HAUPP PASCAL 287	4545	4478	4478	4054	2500	2174	4688	1875	:35
IBM P PASCAL INTERP2	833	791	268	189	31	26	51	31	
IBM P PASCAL2 NATIVE	1053	977	287	198	32	26	52	31	
IBM P PASCAL INTERP4	520	499	96	64	8	8	15	8	32:08
IBM P PASCAL4 NATIVE	593	566	99	65	8	8	15	8	31:49
NCI P PASCAL INTERP2	1114	1058	942	701	78	84	124	82	3:57
NCI P PASCAL2 NATIVE	1585	1468	1245	860	80	86	128	83	(())
NCI P PASCAL INTERP287	1435	1429	1442	1331	650	630	733	584	1:34
NCI P PASCAL 287 NATIVE	9009	8850	8772	7335	832	803	971	727	:21
NCI P PASCAL INTERP4	972	937	615	382	41	45	68	44	6:14
NCI P PASCAL4 NATIVE	1321	1236	725	422	41	46	70	44	(())
NCI P PASCAL 487 INTERP	1312	1322	1305	1281	622	595	695	564	1:40
NCI P PASCAL 487 NATIVE	8380	8000	7481	6803	795	755	934	703	:22

must configure the system to your particular needs. Fortunately, this is much easier than it sounds, takes about 5 - 10 minutes, and requires no programming. Once the appropriate package is constructed, execution becomes completely transparent to the user.

Special care should be taken with the 8087 support packages from both Seattle Computer and Microware because the IBM BASIC Compiler will convert double precision number to single precision in any expression involving both single and double precision numbers. This will result in a loss of accuracy.

Results

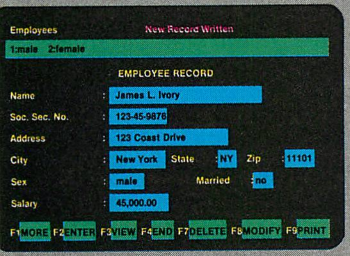
We used two programs to test the performance of the various packages. The first program consisted of a series of FOR loops that performed the four basic arithmetic functions and calculated logs, exponents, square roots, and sines. The amount of time needed to perform the calculations was timed with a stop watch and the number of oper-

DATAEASE™

The Complete System To Organize Your Business The Way You Want It

You can start using it in minutes, and within hours have a complete application.

It is Easy to Use:



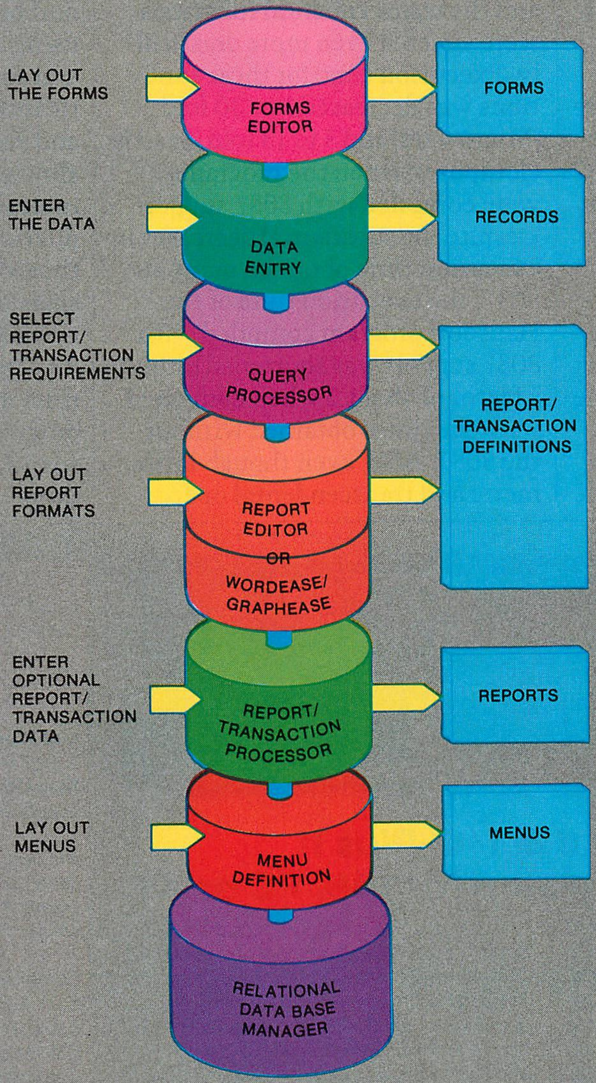
- Fully menu driven to remove the guesswork.
- Full use of function keys with the assignments always displayed. All cursor movement and editing functions on the keyboard are supported.
- Provides full color support and screen style customization.

System Features:

- A list of users can be kept, with passwords and security levels.
- The print style is customizable.
- Data can be interchanged with mainframe computers, other data bases or spreadsheet programs.
- Data base back-up and restore functions are integrated to reduce errors.

System Requirements:

- Available on IBM-PC, DEC Rainbow, TI Professional, Victor 9000, and other 16 bit computers.
- Supports floppy or hard disk drives.
- Supports all popular printers.



It Provides A Complete Range of Features:

- Lay out your forms on the screen. Define data-entry fields anywhere by answering simple questions. Revise the forms any time without losing data.
- Easy, fast and accurate way to fill-in the forms and enter, view or revise records.
- The reports or transactions requirements are defined in the English-like query language by answering simple questions. Select records any number of ways, combine any number of forms per report, perform calculations, get totals, sub-totals, or statistical summary. For complete transaction processing, delete, modify or enter records.
- Reports are automatically formatted in several ways, or specify your own format using the Report Editor.
- To format a report complete with text, data, and graphs; use the optional WORDEASE™ word-processor or the GRAPHEASE™ graphics facility.
- Display or print the report output, or take it to your spreadsheet, word processing, or graphics programs.
- Organize the access to your forms and reports by setting up your own menus.
- The underlying Relational Data Base Manager provides efficient storage and retrieval of records. It uses B-tree indices for efficient accessing of records, and cache memory to retain the most recently used disk sectors to reduce disk access by 50 percent.

These are Some of the DATAEASE Applications:

Inventory Control	Accounting	Medical Office Systems	Social Sciences
Personnel Management	Order Processing	Real Estate	Bibliography
Job Accounting	Market Analysis	Agriculture	Library Management
Mailing List	Financial Portfolio	Distribution Management	School Records

Software Solutions, Inc.

305 Bic Drive, Milford, CT 06460 • (203)877-9268 • (800)243-5123

For information, please write or call. Dealer inquiries welcome.

CIRCLE NO. 142 ON READER SERVICE CARD

ations performed per second was calculated. In order to get an idea of how the packages performed under real world conditions, we ran a program developed by P. Sarnak of the Courant Institute of Mathematical Sciences at New York University and R. Phillips of Stanford University. The program calculates Hausdorf dimensions. While Hausdorf dimensions are admittedly esoteric, the program contains a good balance of arithmetic and control operations and also provides answers that have been verified both by calculation on a mainframe and by theoretical analysis.

The results are shown in Figure 1. The first four lines demonstrate as expected that compiling a BASIC program generally shortens its execution time. However, there were significant differences in the amounts of time saved for different operations. The time needed to execute a double precision division decreased by 91% whereas the time needed to calculate double precision logarithms and exponents actually increased by 26-33%. Execution time for the Hausdorf program decreased by a factor of 6 to 7.

The programs were then compiled with 8087 support from either Microware or Seattle Computer in either single or double precision and rerun. The programs compiled with 8087 support executed the four basic arithmetic operations 5 to 10 fold (Microware) and 1.4 to 2.4 fold (Seattle Computer) more rapidly than did the program compiled without 8087 support.

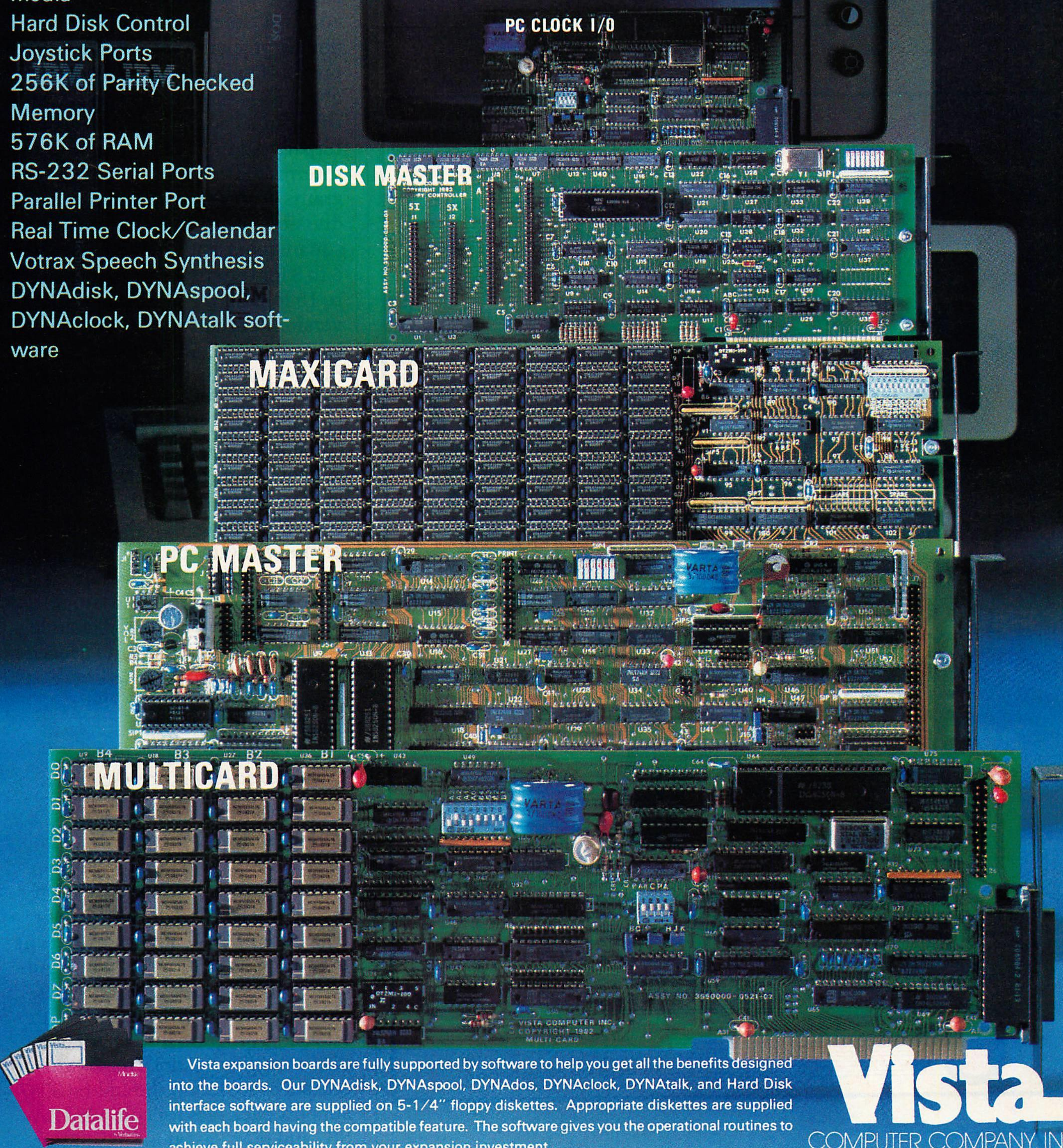
Calculations of square roots and the transcendental functions were affected even more drastically. Execution time for these functions was decreased 5 to 30 fold (Microware) and 3 to 13 fold (Seattle Computer) when the programs were compiled with 8087 support. Despite the substantial decreases in execution time obtained using an 8087, the execution time of the Hausdorf program compiled with 8087 support was shortened by only a factor of about 2 when compared with the times obtained without the 8087. Also, even though arithmetic and transcendental operations executed 2 to 4 times more rapidly with Microware's 8087 support package than with Seattle Computer's package, the times obtained for the Hausdorf program using either the Microware or Seattle Computer packages were almost identical.

When the same programs were rewritten in IBM Pascal and compiled in single precision without 8087 support, the time needed to perform addition, subtraction and multiplication and to run the Hausdorf program equaled that obtained using a compiled BASIC program and an 8087. In contrast, logarithms, exponents, square roots, and sines were calculated very slowly, even more slowly than with interpreted BASIC. However, compilation with 8087 support from either Microware or Hauppauge speeded up execution of these functions dramatically (up to 150 fold for square roots) and also cut the time needed to run the Hausdorf program by 80% to 35 seconds (versus 17.5 minutes needed to run the same computation in BASIC, a net

Expanding Your PC™ or XT™? ASK for Vista

CHECK THESE FEATURES

- Multiple Floppy Disk Control - interface 5 1/4" and 8" media
- Hard Disk Control
- Joystick Ports
- 256K of Parity Checked Memory
- 576K of RAM
- RS-232 Serial Ports
- Parallel Printer Port
- Real Time Clock/Calendar
- Votrax Speech Synthesis
- DYNAdisk, DYNAspool, DYNAclock, DYNAtalk software



Vista expansion boards are fully supported by software to help you get all the benefits designed into the boards. Our DYNAdisk, DYNAspool, DYNAdos, DYNAclock, DYNAtalk, and Hard Disk interface software are supplied on 5-1/4" floppy diskettes. Appropriate diskettes are supplied with each board having the compatible feature. The software gives you the operational routines to achieve full serviceability from your expansion investment.

Your Vista Dealer has more details on our fine family of expansion boards and floppy disk drive systems. See him for a full listing of Vista's economical, high quality products.

For the Vista Dealer or Distributor nearest you - call our Vista Sales lines.

CIRCLE NO. 101 ON READER SERVICE CARD

IBM, PC, XT are trademarks of International Business Machines Corp.

Vista
COMPUTER COMPANY, INC.
1317 East Edinger Avenue
Santa Ana, California 92705
(714) 953-0523 (800) 854-8017
Telex 910-595-1565

increase in execution speed of 30 fold).

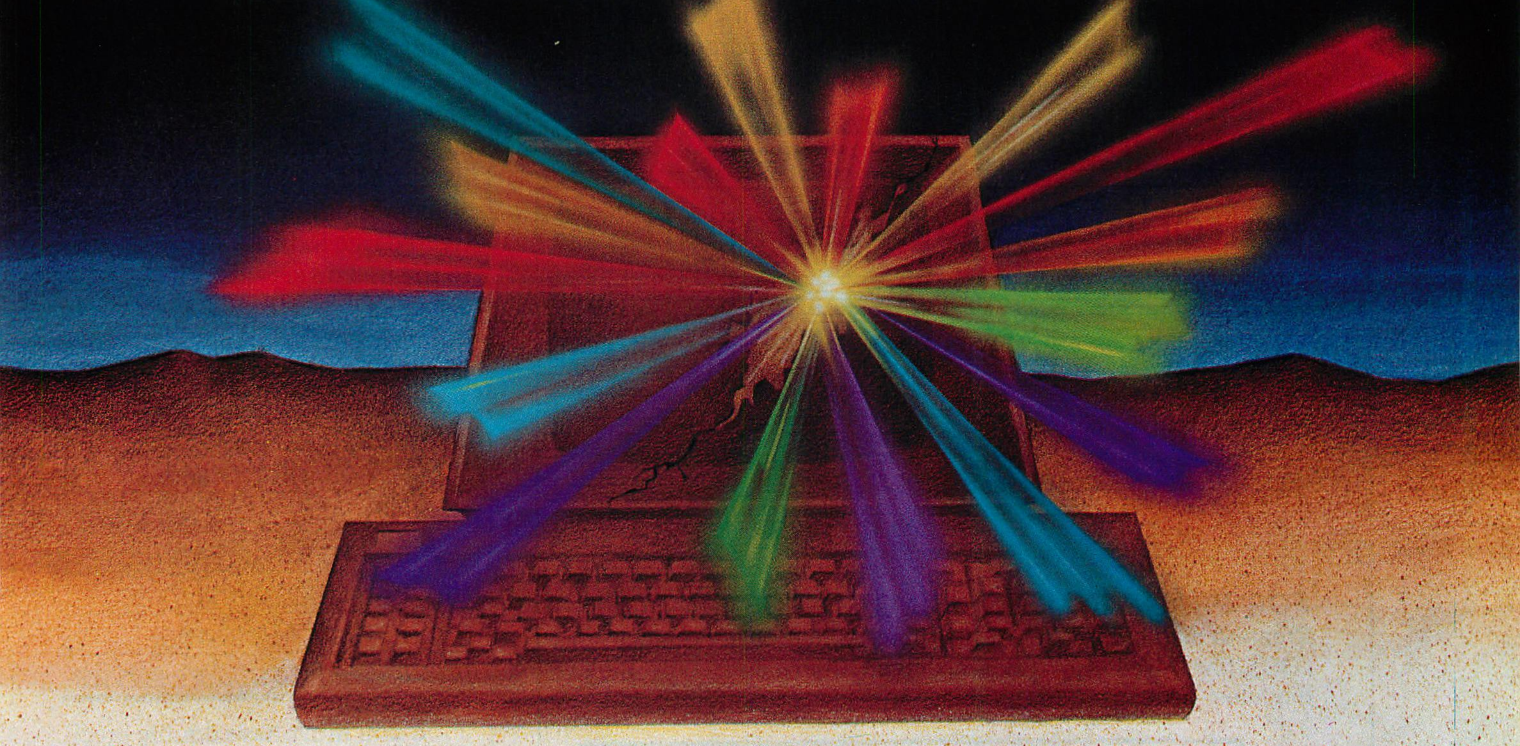
The same programs were then run under IBM's version (version 4.0) of UCSD p-system Pascal. While single and double precision addition and subtraction (either interpreted or compiled to native code) and multiplication and division (interpreted) ran about as quickly as they did under interpreted or compiled BASIC, multiplication and division in the native code versions ran slower by a factor of about 5 to 7.5. Square roots and transcendental functions were also calculated very slowly, taking 2.5 to 30 times as long as with BASIC. Double precision versions of the Hausdorf program also took longer to run in either interpreted (83%) or native code (7.5 fold) than did the equivalent program in compiled BASIC.

Lastly, the two test programs were run under NCI's version (version 4.1) of UCSD p-system Pascal. Interpreted and compiled versions of the program in both single and double precision but without 8087 support performed addition, subtraction, multiplication, and division as rapidly as did the equivalent programs run in compiled BASIC. However, square roots and the transcendental functions were executed 2 to 3 times more slowly in the double precision version. The interpreted Hausdorf program took about 50% more time to execute than did its compiled BASIC counterpart. Adding 8087 support to the interpreted versions of the programs decreased execution time for division, logarithms, exponents, square roots, and sines by a factor of 6 to 8 for the single precision and 10 to 15 fold for the double precision version. Execution time for the Hausdorf program was decreased a further 60% for the single precision version and 73% for the double precision version. Programs that were converted to native code with 8087 support executed much more rapidly than did interpreted programs with 8087 support or compiled programs without 8087 support. The four basic arithmetic operations were executed 6 to 8 fold (single precision) and 6 to 16 fold (double precision) faster. The time needed to run the Hausdorf program decreased by a further 78% for both single and double precision to 21 and 22 seconds respectively, the lowest execution times recorded.

Conclusions

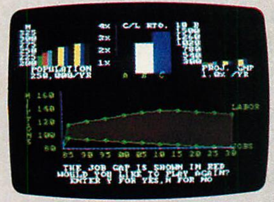
Our data clearly show that compiling the BASIC version of the Hausdorf program increased execution speed by a factor of 6. Subsequent addition of 8087 support further cut execution time by a factor of 2. Addition of 8087 support to IBM Pascal decreased execution time of the Hausdorf program by a factor of 5. Since the price of both hardware and software for all the add-on 8087 support packages for compiled BASIC and IBM Pascal is $395 or less, they are a very good investment if,

COLORPLUS™
Shatters The Mold.



A vibrant, varied new world of color is now at your fingertips.

We've shattered the mold for color graphics adapters with COLORPLUS, a high resolution, full color breakthrough fully compatible with IBM Personal Computer hardware and software.



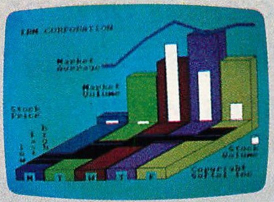
COLORPLUS produces incredibly sharp, detailed multi-color graphics of professional quality at a low PC price. Its state of the art biplanar technology enables high resolution 4 color 80 character graphics or medium resolution 16 color 40 character graphics.

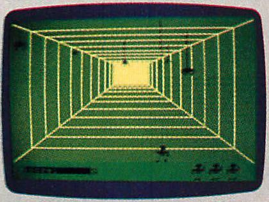
The current IBM PC capabilities of high resolution 1 color 80 character graphics and medium resolution 4 color 40 character graphics simply pale in comparison, now that the era of COLORPLUS has arrived.

XT COMPATIBLE

Using only a single expansion slot, COLORPLUS includes an integrated parallel port, thereby saving you an expansion slot for future needs. No modification is required to existing software in the IBM modes. Enhanced software supporting COLORPLUS Graphics capabilities is currently available from major software suppliers.

For over 31 years, Frederick Electronics Corporation has developed high quality electronic equipment world-wide. The key to our success is our international reputation for customer support. PC+ products will continue in this tradition. Frederick Electronics Corporation is a subsidiary of Plantronics, Inc., a publicly owned company listed on the NYSE as PLX.





The goal of PC+ products is to expand your options, to make your personal computer more valuable and productive. COLORPLUS is just one example of our commitment to that objective.

Contact us for the name of your nearest COLORPLUS dealer.

PC+™ PRODUCTS

PLANTRONICS™
Frederick Electronics
7630 Hayward Road, P.O. Box 502
Frederick, Maryland 21701
Telephone: 1-800-638-6211
(301) 662-5901, TELEX: 893438

CIRCLE NO. 368 ON READER SERVICE CARD

PC+ and COLORPLUS are trademarks of Plantronics/Frederick Electronics Corp. IBM is a registered trademark of International Business Machines Corp.

1) you do enough numerical computations to keep your machine occupied for more than a few minutes and 2) your software will support the 8087 chip. One should keep in mind, however, that a BASIC compiler or an IBM Pascal compiler (at $300 each) is also required. In terms of marginal price/performance ratio you can increase your PC's computational ability 5 to 13 fold for less than a 20% increase in your initial cash outlay (BASIC or IBM Pascal compiler - $300, 8087 plus software support package - $295-$395).

If you plan to use BASIC, either the Microware or the Seattle Computer packages support double precision numbers, and the prices for both are $375 and $395 respectively. The Microware package is preferable because it executes the basic operations more rapidly and Microware provides a free update service.

If you use IBM DOS Pascal, the package from Hauppauge Computer Works looks like a better buy than the package from Microware. It is cheaper ($295 vs. $375), it provides a carrier for the 8087 chip which makes installation easier, and an extra 8088 chip is provided in case you happen to own a PC with an early version 8088 which won't work with an 8087. Hauppauge offers a software update service for $30/yr; they state that 8087 support for IBM DOS BASIC and IBM DOS FORTRAN will be available later in the year and will be delivered free to those who subscribe to the update service. The negative aspect of using IBM DOS Pascal for numeric computations is that the language only supports single precision real numbers.

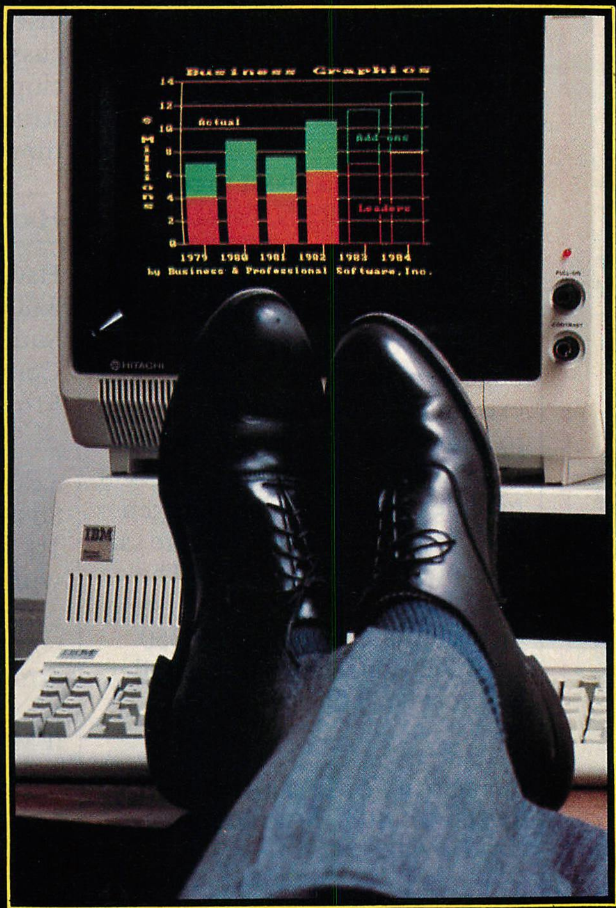
The Microware package is preferable because it executes the basic operations more rapidly, and Microware provides a free update service.

If you must have double precision and rapid execution, then NCI's UCSD p-system Pascal, which, with an 8087, runs the Hausdorf program in double precision about 4 to 5 times as fast as BASIC and faster than all the other packages tested, is probably what you need. While the NCI UCSD p-system Pascal package costs a good deal more ($845) than other compilers, it is an easy-to-use, integrated program development system which comes with a variety of utilities and an editor. The total cost of an NCI UCSD Pascal p-system package with an 8087 chip is $1070 ($845 plus $225 for an 8087) versus about $595-695 for either IBM DOS BASIC or IBM DOS Pascal.

Great Expectations

Several recent developments deserve mention. IBM is now selling a chip set that contains an 8088 chip and an 8087 chip for $260. At present, the only IBM software product

Relax.



BPS BUSINESS GRAPHICS™

for the IBM® PC

Seeing answers without staring at numbers.

Discover what your data really means.

With BPS Business Graphics you can analyze your data better by charting it. You can see trends, compare performance to projections, and find patterns that printouts often hide. The program lets you decide what your graphs should look like—and you can customize and store the formats you use most often.

Chart the data you already have.

You can graph data directly from the programs you already use, including VisiCalc®, 1-2-3™, Multiplan™, and dBASE II™—even data retrieved from your mainframe—and you won't have to retype a single number. You can manipulate the data even while you're graphing it with powerful statistical functions such as curve-fitting and moving averages.

See the answers instantly.

BPS Business Graphics quickly gives you professional-looking graphs in a wide variety of formats, so you can make colorful presentations on your monitor. And since the program works with more than sixty popular printers and plotters, you can get high-quality paper copies of your graphs, and even overhead transparencies.

For information on business graphics for the Apple, IBM PC, and Wang PC, call us at (800) DIALBPS or (617) 491-3377.

Business & Professional Software, Inc. 143 Binney Street Cambridge, Massachusetts 02142 USA

CIRCLE NO. 119 ON READER SERVICE CARD

that supports the 8087 is the new IBM implementation of the APL language, which costs $195.

New Microsoft releases of the FORTRAN and Pascal compilers are expected. Both these compilers will generate in-line 8087 code and support double precision real numbers. IBM will presumably release these products as IBM products in the near future.

Microware is about to release the Intel FORTRAN, Pascal and PLM86 compilers, which run under the RTOS operating system. RTOS supports the full memory space of the IBM PC, allowing programs as large as approximately 640K to be executed. Utilities will be provided to allow runtime access from IBM DOS to previously compiled RTOS modules. All these compilers generate in-line 8087 code, and are said to be optimizing compilers. The cost of the RTOS operating system including one language is $1350. Each additional language compiler will cost $675.■

Directory of Literature and Reviewed 8087 Software and Hardware

Literature

Robert J. Brady & Co.
Bowie, MD 20715

8087 Applications & Programming for the IBM-PC and Other Personal Computers. R. Startz. Deals with assembly language 8087 programming.

BYTE/McGraw-Hill
70 Main Street
Peterborough, NH 03458

8086/8088 16-Bit Microprocessor Primer. Christopher Morgan and Mitchell Waite. $16.95. Not nearly as detailed as the Intel reference regarding the 8087 but well written with good diagrams.

Intel Corporation
Literature Department
3065 Bowers Avenue
Santa Clara, CA 95051
408-987-8080

"Getting Started with the Numeric Data Processor". No charge. Order # AFN-01300B-1 (AP-113).

iAPX86,88 User's Manual. $10.00. Order # 210201. A detailed discussion of the 8087. Also discusses the 8088, 8086, and other chips in detail. Includes the full instruction set of the 8087 along with timing information.

Hardware (8087 Chips)

Hauppauge Computer Works
358 Veterans Memorial Highway, Suite MSI
Commack, NY 11725
516-360-3827

Price for a single 8087 (comes in a carrier along with a compatible 8088 if you happen to be one of the 1% with non-compatible 8088s) without software is $225. Hauppauge also sells 8087 software support packages for IBM Pascal and assembly language - see below.

IBM

Is now selling 8087s (along with a cardboard heat shield and an 8088) for $260 in order to support their recently announced APL for the PC. While somewhat more expensive than the other vendors, the chip (marked with their initials) if purchased from IBM will be covered under their service contracts.

MicroWare
P.O. Box 79
Kingston, MA 02364
617-746-7341

Price for a single 8087 without software is $223. Contact them for quantity pricing. MicroWare also sells 8087 software support packages for various languages - see below.

Software

Hauppauge Computer Works

Hauppauge sells a package consisting of an 8087 in a carrier, an 8088, and software support for IBM Pascal and Macro-Assembler. This package costs $295. Software up-

dates for 1 year are $30. The vendor states that software support for BASIC and FORTRAN will be available later in the year and will be sent to update service subscribers. Hauppauge gives discounts on volume purchases to groups; inquire directly.

MicroWare

MicroWare sells 8087 support packages for compiled BASIC, IBM Pascal, IBM FORTRAN, and the MacroAssembler. A package consisting of an 8087, a publication (87/88 Guide), and any one language support package costs $375. Additional language support packages cost $150 each. The chip is warrantied for 180 days except for damage to the pins during installation. MicroWare gives discounts on large purchases to either groups or businesses; inquire directly.

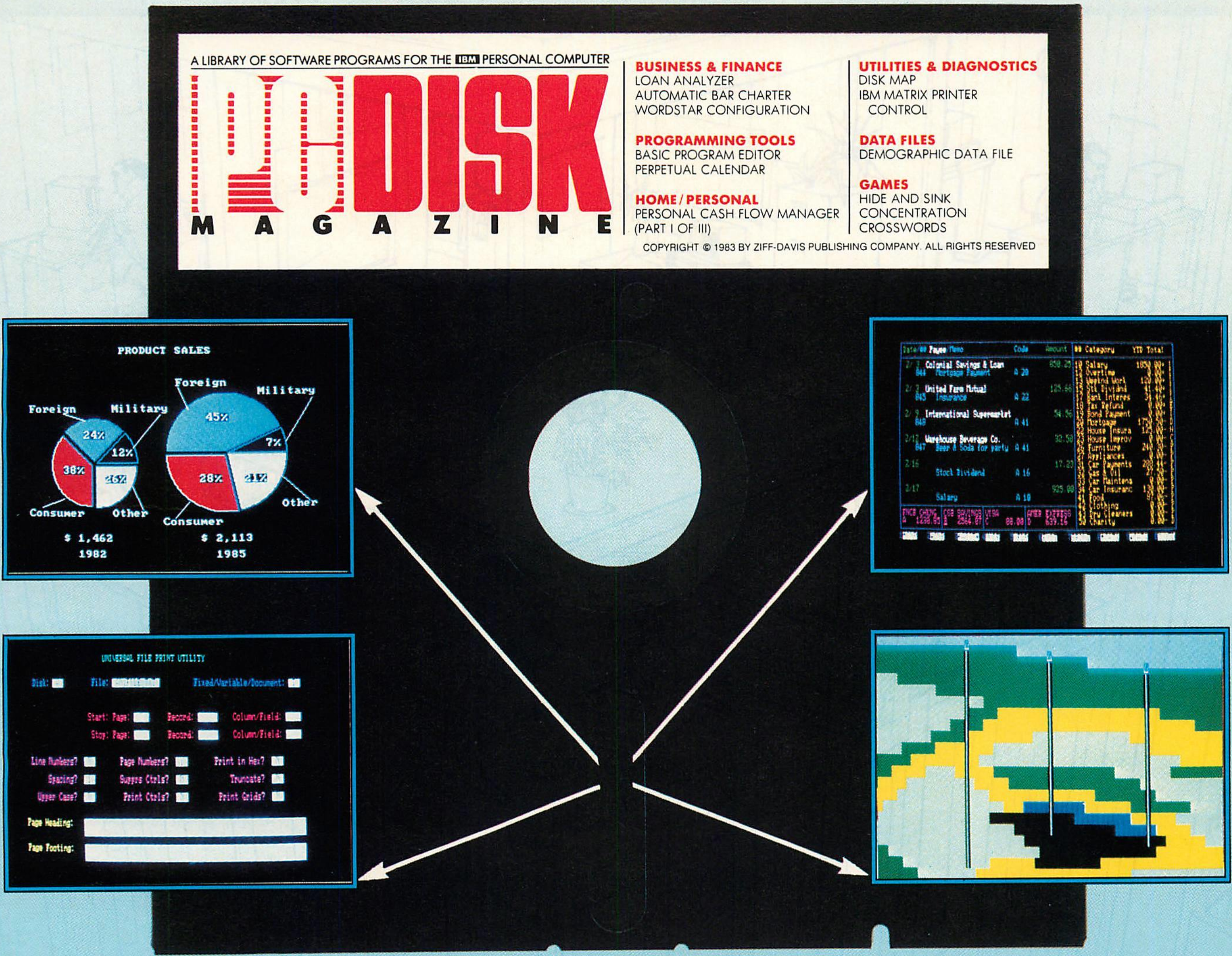
Network Consulting, Inc.
Discovery Park
Suite 110-3700 Gilmore Way
Burnaby, B.C.
Canada V5G 4M1
604-430-3466

NCI sells a program development system which supports the UCSD p-system. The package includes an editor, Pascal compiler and native code generator, 8087 support, filer, and numerous other utilities. The entire package costs $845. NCI also sells 8087s ($350).

Seattle Computer Products
1114 Industry Drive
Seattle, WA 98188
800-426-8936

Seattle sells a package consisting of an 8087 chip, and software support packages for both compiled BASIC and MacroAssembler. Neither the 8087 nor the software are sold separately. The price for the package is $395. The 8087 is warrantied for 90 days. ■

CIRCLE NO. 378 ON READERS SERVICE CARD



PRODUCT SPECIFICATIONS

- Programs will run on IBM personal computers under PC-DOS and MS-DOS versions 1.1 and 2.0 and require a minimum of 64K.
- Most programs will be written to run on both monochrome and color displays; however, some games and utilities may be specifically developed for color display.
- Most programs will be written in advanced Basic; however, some machine language and compiled code may be used.
- Programs and documentation are copyrighted by Ziff-Davis Publishing Company. All rights of reproduction in all forms and media strictly reserved.

PROGRAM SUBMISSIONS

If you wish to submit a program for inclusion in future issues, please write to: PC DISK MAGAZINE, Attn: Editor, One Park Avenue — Dept. 732, New York, N.Y. 10016.

GUARANTEE

- All programs are fully tested and guaranteed to run. Damaged or faulty disks will be replaced at no charge.
- If you wish to cancel a subscription, simply return the most recent disk in its sealed package and you will receive a full refund for this copy and on all unmailed issues.

Save $60 With This Introductory Savings Offer. Get 6 Big Months of PC DISK MAGAZINE And Save 33%.

PC DISK MAGAZINE P.O. Box 596, Morris Plains, N.J. 07950 **8H00**

YES, please start my subscription to PC DISK MAGAZINE for 6 issues at the Special Introductory Price of just $119 (less than $20 per issue, as little as $2 per program). I save $60 — 33% off the full price of $179.

Check one: ☐ Please bill me. ☐ Payment enclosed.
☐ Please charge to my credit card.
☐ American Express ☐ Visa
☐ MasterCard

Credit Card No. ____________ Exp. Date ____________

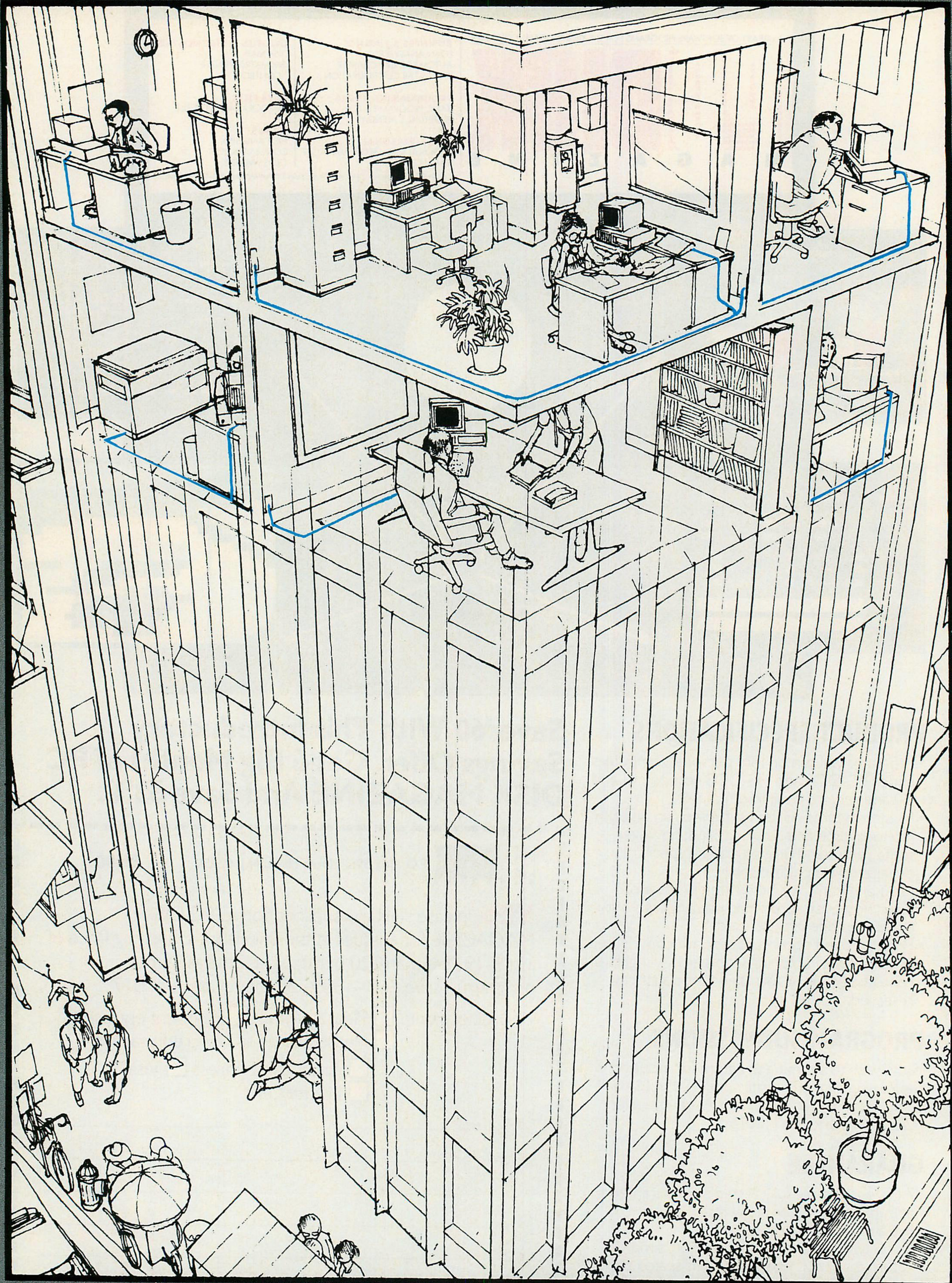
Cardholder Name ____________

Name ____________

Address ____________

City ____________ State ____________ Zip ____________

MAIL NO-RISK COUPON TODAY – AND SAVE 33%!



PCnet- for the IBM PC Low Cost Networking

How one low-cost local area networking system works with the IBM PC.

Bruce W. Churchill

As much publicity as the IBM PC has received over the past year, one would be led to believe that the excitement would soon end. In fact, the excitement may be just beginning, because the real promise of the PC is not the machine itself, but its potential usage. Of equal significance to the arrival of Big Blue's entry into the microcomputer sweepstakes is the recent surge in interest in local area networking. In almost every business, communication does or will play a major role in determining success or mediocrity. It is a generally accepted fact that more than three-fourths of all business communications occur in the local environment. The IBM PC and local area networks form a natural partnership—one that will become increasingly familiar in the months ahead.

One of the first networks available for the PC, PCnet by Orchid Technology, provides a relatively simple, low cost implementation of resource sharing and intercommunication among micros. This article will introduce the reader to the strengths and limitations of PCnet and will explore some typical applications for this technology.

The IBM PC and local area networks form a natural partnership, one that will become increasingly familiar in the months ahead.

PCnet System Overview

PCnet was designed specifically for the IBM PC and its native operating system, PC DOS. It is classified as a distributed bus, baseband local network with CSMA/CD access protocol. The network is capable of addressing up to 64,000 devices and can therefore physically accomodate almost any conceivable network that might be found in practice. PCnet also supports IBM PC plug-compatible computers, and the MS DOS cousin of the PC DOS

Bruce W. Churchill, a captain in the United States Navy, is head of tactical command and control systems, U.S.N.

operating system. Examples of this compatibility are the COMPAQ, Colby, Columbia MPC, and Eagle 1600 computers. It is compatible with PC DOS version 1.1 and is currently being modified to work with version 2.0. PCnet software interfaces directly with PC DOS and makes the network look like another Input/Output device to the operating system. There are no servers required on this network. Instead, shared printers and hard disk devices are connected directly to designated PCs. PCnet will support any plug-compatible peripherals that are supported by PC DOS.

Hooking up the cable is almost as easy as plugging in the power.

PCnet Hardware The PCNet hardware consists of a single network adapter card that plugs into one of the available expansion slots on each IBM PC in the network. On the back of this board is a BNC-type connector, which plugs directly into the network cable via an inexpensive T-connector, as shown in figure 1. The network cable is a standard 75-ohm CATV type coaxial cable. It runs end-to-end throughout the area where the networked PCs are installed. Depending on cable rating, wire runs of up to 7,000 feet are possible, allowing an entire work area to be pre-wired and PCs to be added to the network or easily moved. Hooking up the cable is almost as easy as plugging in the power. The PCNet adapter card contains proprietary CSMA/CD logic to control access to the network by attached PCs. A nominal serial data rate of 1 megabit/sec provides high speed operation and optimizes direct memory access operations with the PCs in the network.

PCnet Software Each PCnet adapter card comes with software that integrates the network into PC DOS. This DOS interface is designed so that standard single-user PC applications can use the network without change. More advanced multi-user systems can be developed from single-user applications software with varying degrees of program modification. The four main features of the DOS interface are disk sharing, file locking, remote execution, and datagram interfacing.

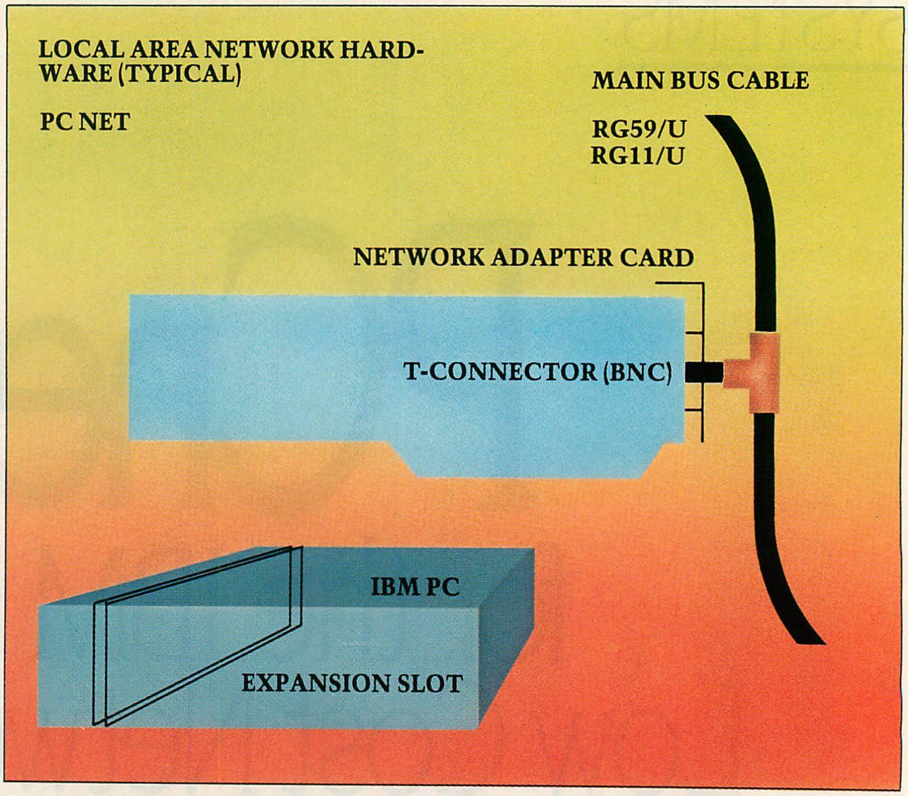


Figure 1: The BNC connector plugs directly into the network cable.

Disk sharing allows PCs to share hard disks or floppy drives installed in other PCs.

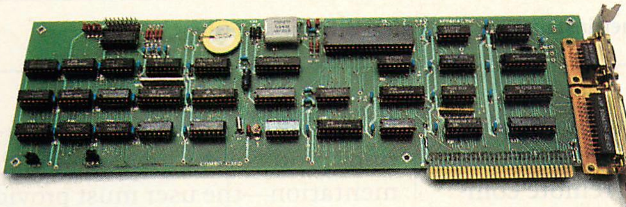
Disk Sharing Disk sharing allows PCs to share hard disks or floppy drives installed in other PCs. The shared disks appear to be locally attached. For example, if a PC has one diskette drive and shares a hard disk via the network, the user can access the local floppy as Drive A:, and the shared disk as Drive B:, C:, or D: as shown in figure 2. Drives B:, C:, and D: then become volumes on the shared hard disk for this particular example. Any future reference to B:filespec, C:filespec or D:filespec while loading or saving programs or accessing data files will activate the hard disk as if it had been attached to the requesting PC. The assignment of drive designations (A: or B:, for example) to local and shared drives is completely flexible so that a user's Drive "A:" does not have to correspond to their local floppy drive. Figure 2 shows a typical PCnet installation and logical drive relationships. The disk sharing interface has the inherent advantage of being able to work with any hard disk that is compatible with the IBM PC, whether internally or externally mounted.

File locking In order to implement a multi-user, multiple-PC application sharing common data files, there must be a mechanism to ensure that these files are only accessed by one user at a time. The denial of access to a file that has been opened is known as file locking. Access denial can be specified at the file or the record level. The

THE ONE AND ONLY

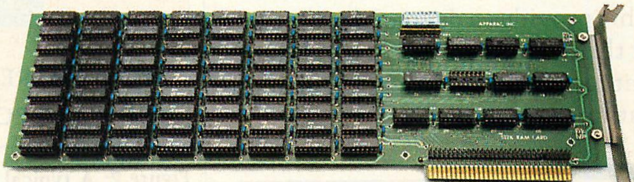
CRAMBO

You've got a problem. Go to your favorite computer store. Ask to see their Combo Cards. Then ask about their RAM Cards. See the problem? Just too many to pick from. Now, ask to see the CRAMBO™. No problem. There's only one. Ours. Except you don't know what a CRAMBO is. You've probably guessed it's a Combo Card piggy backed to a RAM Card. So it only takes up one slot in your IBM/PC. That's right. But here are the details.



THE COMBO II CARD:
For **$189** you'll get a Clock Calendar, Async Communications, Parallel Printer and a Game Adapter.

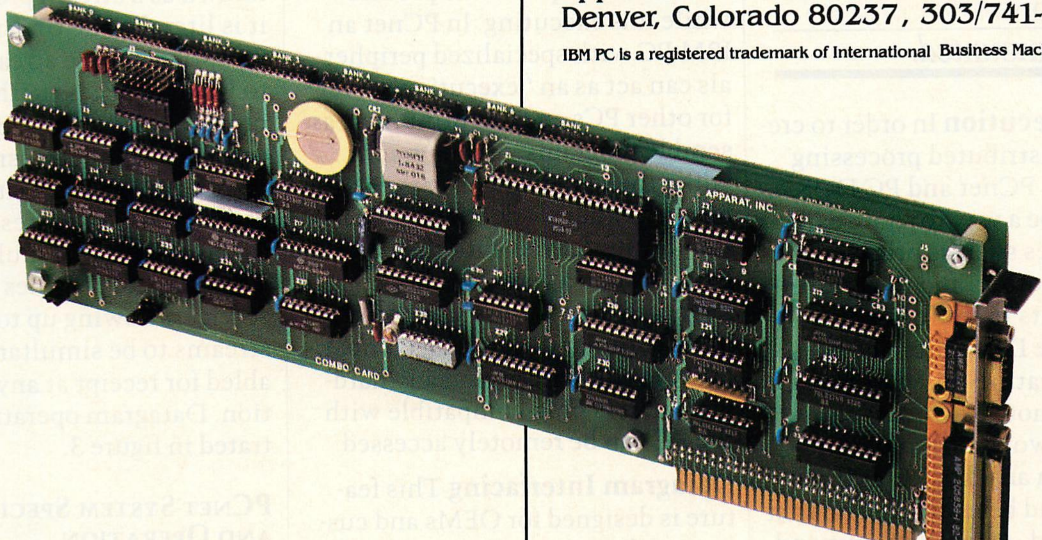
THE 512K RAM CARD:
It too, is only **$189** with 64K of RAM installed. And when you need more RAM, 64K increments are available for $64 each. And, SDRIVE, the electronic disk emulator, is available at no cost with the RAM Card.



You can buy either of our boards separately for use in the XT or PC expansion chassis. But only our boards can be piggy backed to give you the one and only CRAMBO. And for only **$359.**

Go check out the CRAMBO at the same store carrying all those Combo Cards and RAM Cards. If they don't have the CRAMBO have them call us. **800/525-7674.** Or write: Apparat, Inc. 4401 South Tamarac Parkway, Denver, Colorado 80237, 303/741-1778.

IBM PC is a registered trademark of International Business Machine Corp.



Apparat, Inc.

CIRCLE NO. 106 ON READER SERVICE CARD

term unlock refers to the removal of the access denial when the file is closed. Because PC DOS does not support file locking (remember that it is a single-user system), the DOS interface includes function calls to lock and unlock files by name or by name and record number. These assembly language function calls can be used from within an application program to provide the required file access protection. If two or more PCs try to modify a common file at the same time, one will simply wait until the other is finished with its access before the proceeding. This is essentially the same principle used in installations running multi-user operating systems. Another means of file locking is by the use of three supplied utility programs that can be executed from a batch file. This provides a degree of protection at the operating system level without requiring modification of applications programs.

The disk sharing interface has the inherent advantage of being able to work with any hard disk that is compatible with the IBM PC, whether internally or externally mounted.

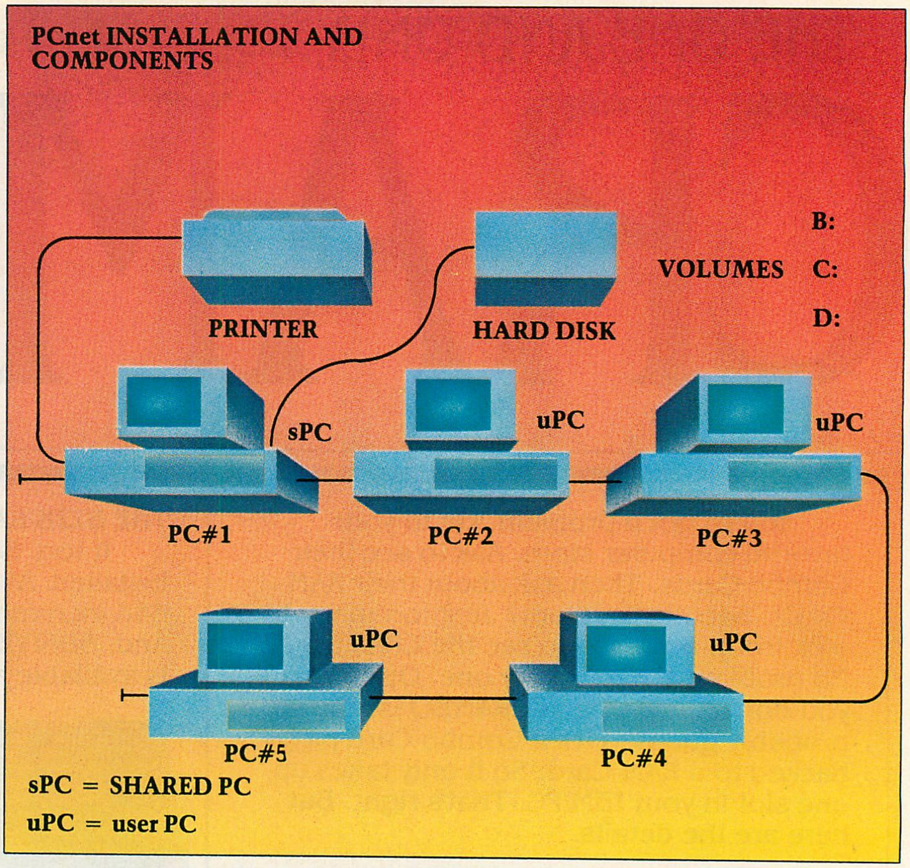


Figure 2: A typical PCnet installation and logical drive relationships.

Remote Execution In order to create a truly distributed processing system with PCnet and PC DOS, there must be a way to share expensive resources such as printers, modems, and plotters and to perform what amounts to multi-tasking operations. The DOS interface does this with a feature called remote execution. Remote execution allows a user on one workstation to run a command on another PC as if the command had been entered on that PCs keyboard. After the command executes, the remote user is informed by a short "beep" on his or her PC. In modem network systems, this feature is roughly analogous to a host communications program in which the host computer can be controlled by remote commands coming into its modem. The analogy is not complete because the PCnet remote execution feature does not provide the requesting user video output of the process while it is executing. In PCnet an IBM PC with specialized peripherals can act as an "execution server" for other PCs and can print files, assemble programs, or plot graphs while the requesting user is performing other work. Remote execution is interleaved with the host DOS in such a way that the keyboard takes priority over the remote command. Software compatibility is guaranteed, and any hardware that is plug-compatible with the PC can be remotely accessed.

Datagram Interfacing This feature is designed for OEMs and custom application programmers who desire to bypass or supplement the normal network interface with PC DOS. The datagram communications protocol is a low-level implementation—the user must provide desired higher level services such as end-to-end reliability, preservation of message sequence and session management. This feature is also known as a stream protocol in that it is literally the transmission and reception of byte streams from PC to PC. Full access to the CSMA/CD protocol is provided for detecting collisions and retransmitting datagrams as required. Datagrams can be individually addressed or broadcast to designated multiple PCs. The protocol provides multiplexing service, allowing up to three streams to be simultaneously enabled for receipt at any workstation. Datagram operation is illustrated in figure 3.

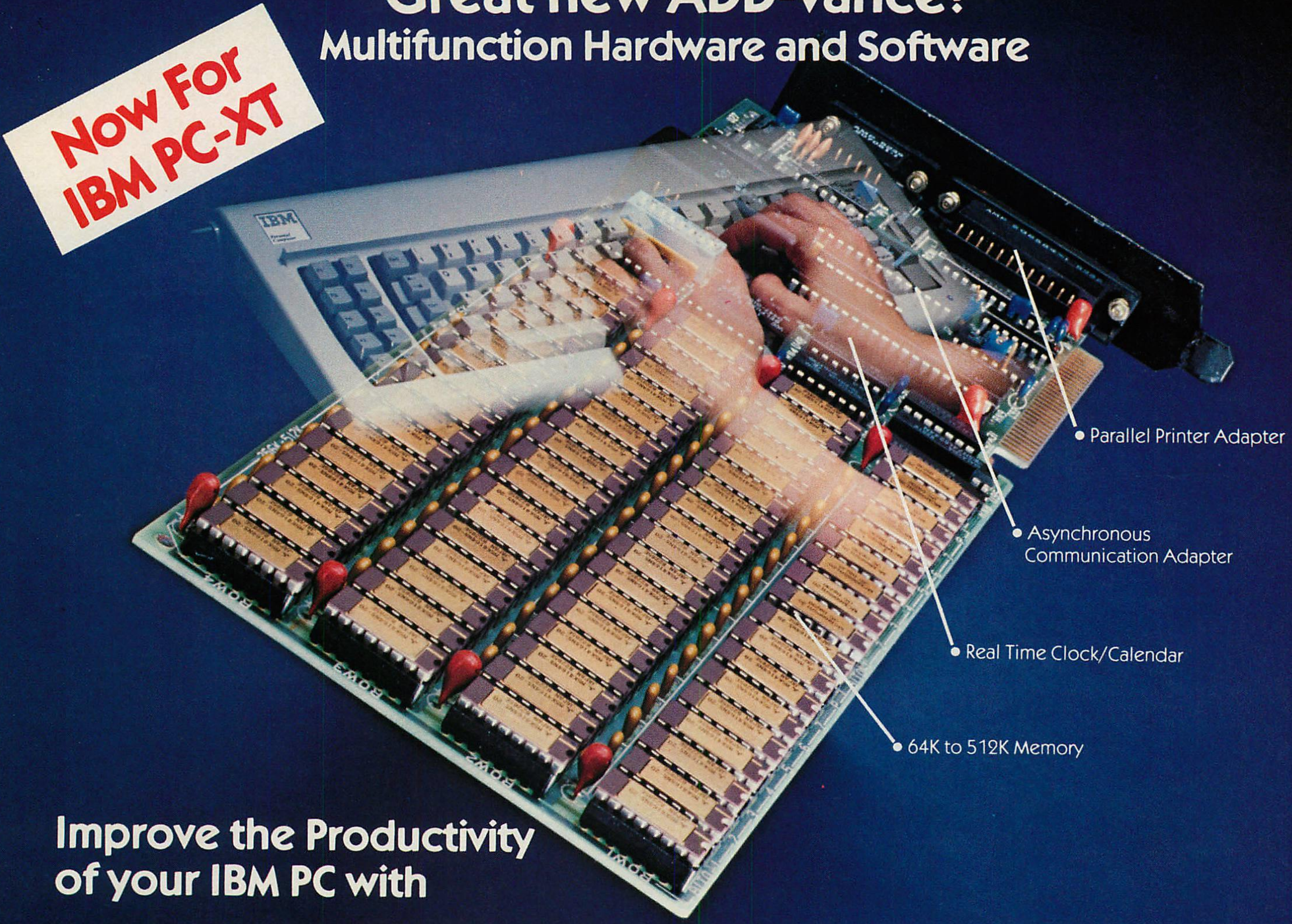
PCnet System Specifications and Operation

Hardware specifications The hardware adapter card communicates with the PC's system board through the use of two interrupt

CIRCLE NO. 170 ON READER SERVICE CARD

Great new ADD-vance!

Multifunction Hardware and Software



Improve the Productivity of your IBM PC with

ADDRAM ELITE

with Ramdisk, Ramspool and Tascmaster Software

Make the most of your IBM PC with this multifunction board which provides an **asynchronous communication adapter, real time clock/calendar, parallel printer adapter, and up to 512k memory with parity.**

ADDRAM ELITE INCLUDES SOFTWARE permitting the real time clock/calendar to automatically set date and time of day in the IBM PC.

INCLUDES RAMDISK SOFTWARE which offers up to 50 times faster speed in operation. Permits utilization of 1 or 2 RAMDISKS, each with up to 320K capacity. Enables user to swap drive C: and D: for A: and B:. Maintains an electronic disc in memory even when you do a re-boot.

INCLUDES RAMSPOOL SOFTWARE which allows the use of the computer while printing information via printer buffer. This applies to either parallel or serial printers. And...more than one spooler can work concurrently.

INCLUDES TASCMASTER SOFTWARE which allows up to 9 tasks to be run concurrently in background mode. Offers the option of running each task on its own Ramdisk or on a disc drive available to the user. Allows the user to move freely from task to task without the necessity of terminating any one of them.

OFFERS OPTIONAL MULTI-TASCMASTER SOFTWARE which allows the addition of multi-user capability to the IBM PC by connecting additional terminals to the serial ports.

OFFERS 64K to 512K MEMORY which supports full IBM PC parity operation and adds great versatility while saving system expansion slots. Board is socketed for easy memory expansion up to 512K by purchase of additional memory chips.

OFFERS ON THE SAME BOARD parallel printer adapter, asynchronous communication adapter and real time clock/calendar.

INSTALLED SIMPLY AND QUICKLY by plugging directly into any one of the expansion slots provided on the IBM Personal Computer, to which it is hardware and software compatible.

TASCMASTER Software
Lets you do several things simultaneously!

Available through local personal computer dealers

Profit Systems, Inc. P.O. Box 1039 Berkley, MI 48072 • (313) 559-0444
Dealer Inquires Welcome

CIRCLE NO. 189 ON READER SERVICE CARD

lines, a single DMA channel, and I/O ports. The adapter card uses lines, channels, and ports not used by any other peripherals in order to avoid contention with devices such as video refresh, floppy drives, and hard disks. These devices use three of the four available DMA channels on the PC system board. One result of this allocation is that PCnet DMA access can only be made to the lowest 64K or RAM, although the impact of this restriction is not significant from a practical standpoint. Maximum cable length is determined by signal loss; for RG-59B/U coax the maximum length is 3,000 feet. The heavier RG-11/U coax will support a cable length of up to 7,000 feet, but is more difficult to install, Cable taps and terminators are standard BNC hardware, available at any electronics supply store. There are no restrictions on where expansion PCs can be tapped into the main bus cable. Again referring to figure 1, the T-connector tap on the main bus cable is designed to attach directly to the BNC connector on the adapter card at the PC back panel.

Software Operation PCs in a network must be initialized as either shared PCs or user PCs (contracted to sPC and uPC respectively). Each sPC and disk drive volume is assigned a name not exceeding 14 characters in length to simplify command and software references. There are some restrictions in the sharing features of the PC software. In general, user PCs can access shared PCs, but neither user PCs nor shared PCs can access users. This is a reasonable restriction, given the design of this network, in

To create a truly distributed processing system with PCnet and PC DOS, there must be a way to share expensive resources.

that user PCs should not be arbitrarily interfered with. Also, an sPC may not directly access another sPC; however, this is a temporary restriction which the manufacturer plans to remove in later versions. If there are multiple sPCs, a uPC with access privileges to appropriate sPCs can copy files between any two of them, in effect acting as a go-between.

Initializing the network software for PCnet consists of the following steps:

Set up sPC workstations

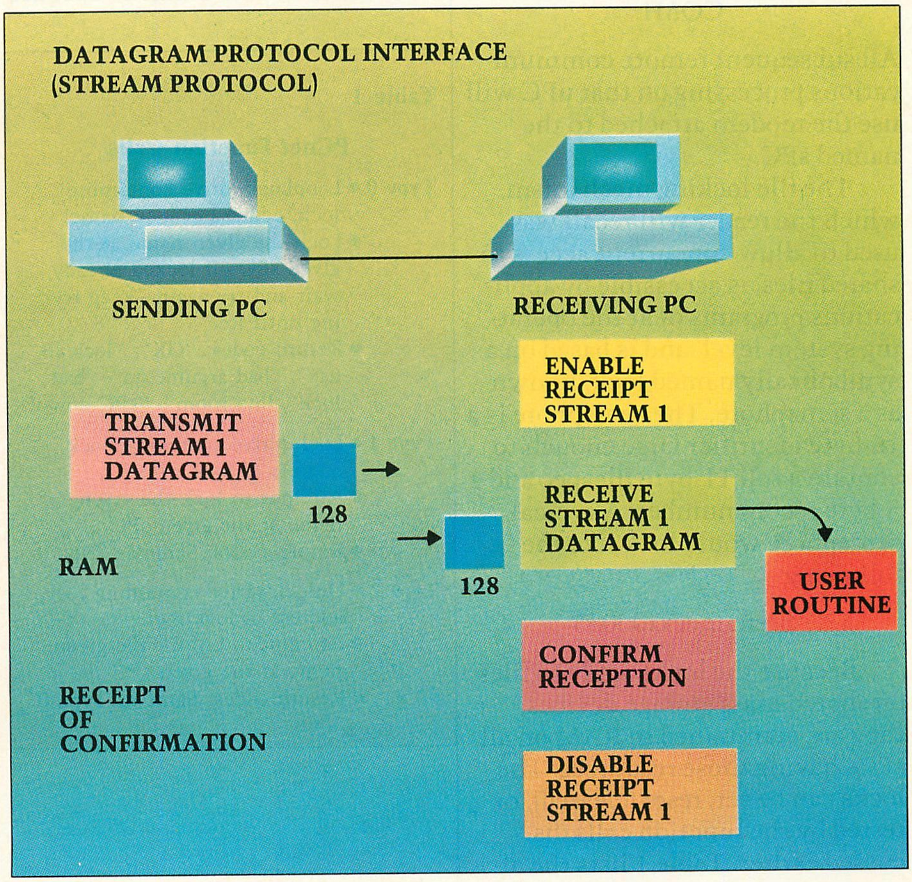
- Create a bootable floppy diskette for the particular hard disk installed
- Copy the files INSTALL.EXE and BIOTAB.COM from the sPC distribution diskette to the boot diskette
- Run INSTALL and reboot the disk
- Copy all files from the sPC diskette to the boot diskette
- Reboot the disk and run the program SPCINST. This allows the user to name shared disk volumes and to set up uPC access permissions. SPCINST can be run at a later time to modify or add permissable uPC accesses

Set up uPC workstations

- Same general flow as for sPCs except that the uPC distribution diskette is used and the program UPCINST is run after the second reboot. The user is asked to name the

In PCnet an IBM PC with specialized peripherals can print files, assemble programs, or plot graphs while the requesting user is performing other work.

Figure 3: Datagram operation is handled at a very low level in the overall network scheme.



sPCs to which access is desired. The desired mapping of drive designators to network shared disk volumes is also entered

Set up Remote Execution on each sPC

- Run the program REINST on the sPCs boot floppy diskette. This program modifies COMMAND.COM to call a PCnet function instead of a DOS function when a command line is received from other than the local keyboard. Local keyboard input has priority

In order to use a shared printer, a uPC user enters the command,

ASSIGN LPT1 TO sPC_name LPT1

All subsequent commands to print a file on the local device LPT1: are then executed on the shared printer attached to an sPC (PC#1 in figure 2). Likewise, a shared modem can be set up on a uPC workstation with the command,

ASSIGN COM1 TO sPC_name COM1

All subsequent remote communications processing on that uPC will use the modem attached to the named sPC.

The file locking mechanism, which the reader will recall was used to allow concurrent access to shared files, is accessible by applications programs or at the operating system level, and is based on a symbolically named lock, known as a semaphore. The semaphore is a 16-byte identifier large enough to contain a full 11-byte filespec and a 5-byte record number. A typical semaphore would look like the following:

filenameext00055

Because the locks apply to files on shared mass storage devices, they are maintained in RAM on all sPCs having those resources. The locks can be set, reset (cleared), or tested by the function calls discussed earlier. Table 1 lists the allowed function calls with the activity performed and the possible results of the call. Certain restrictions are applicable to the use of this concurrent access technique: if any user program creates a new file or changes the length of an existing file, the entire volume containing that file must be closed to multiple access. The use of the full 16-byte semaphore to lock a record will work on indexed files if the 5-digit "record number" is considered to be the first five characters of a designated key field. A planned enhancement is the increase in semaphore size to accommodate larger or multiple index keys.

File access control at the PC DOS level is through the utility programs LLOCK.COM, LOCK.COM, and UNLOCK.COM. These programs have the same effect as (and use) the internal function calls of the same name. They are designed to be used from batch files, as in the following example, using DBASE II and a common access data file—RECORDS.DBF:

The remote execution mode of PCnet allows a uPC user to execute a DOS command on an sPC provided that user is placed on the sPC's access list. The remote command line looks like the following:

re [sPC_name] ⟨command string⟩

The sPC name is optional, and if specified, sends the command string to the designated sPC for execution. Otherwise, the destination sPC is determined by the network software using the list of sPCs with their volume assignments. If one volume is designated a network default volume during system startup, it would execute all commands for which no sPC name was indicated. For example, consider the following DOS COPY command typed on a uPC:

re copy b:filename1 c:filename2

In this case, assume that drives B: and C: are assigned to the shared hard disk. If the network default volume were initialized as C:, the command would execute on the sPC containing volume C: because no sPC name was included in the command. File locking against multiple access is preserved on remotely executed commands so that a user cannot use the system to bypass volume or file protection.

There are distinct limitations to this scheme. Any remote command issued must terminate with a return to DOS in order to work properly. For example, a command to run VisiCalc will not work with

If any user program creates a new file or changes the length of an existing file, the entire volume containing that file must be closed to multiple access.

Table 1

PCnet Function Calls

Type 0	• LoopLock (drive containing lock, address of lock name) • Lock the given name at the given sPC—if locked already, wait and try again—keep trying until free • Return codes: "OK", "lock in use", "bad arguments", "bad lock", "no room", "sPC down"
Type 1	• Lock (drive containing lock, address of lock name) • Attempt to lock the given name at the given sPC • Return codes: Same as Type 0
Type 2	• Unlock (drive containing lock, address of lock name) • Attempt to unlock the given name at the given sPC • Return codes: Same as Type 0

The UCSD p-System. With Speed.

The new NCI implementation of the p-System. For IBM personal computers and compatibles.

NCI now makes the UCSD p-System run 1½ to 5 times faster on the IBM Personal Computer, including the new XT model and compatibles like Corona, Columbia, Compaq, Colby, Dot, Hyperion and Victor 9000. This speed is possible only with our new interpreter.

NCI includes more features for your PC.

NCI offers a wide range of software and hardware support that lets you tailor the p-System to your needs, including:

☐ HARD DISK SUPPORT compatible with most hard disk suppliers, including IBM Personal Computer XT, Corona, Davong, Corvus-Systems and Tallgrass Technologies.

☐ AUTOMATIC RAMdisk SUPPORT (up to 512K) stores files for much faster access.

☐ PRINT BUFFER (up to 64K) eliminates printing bottlenecks.

☐ 8087 SUPPORT speeds numeric and graphics applications.

☐ GRAPHICS SUPPORT with much faster Turtlegraphics software or Tektronix emulation for business and scientific applications.

☐ COMMUNICATIONS SUPPORT LIBRARY links computers across continents.

☐ NCI CUSTOMER SERVICE – we're there when you need us to support the most reliable p-System ever developed for the IBM PC.

For all the advantages of the NCI version of the UCSD p-System call or write:

Network Consulting Inc.
Discovery Park, Suite 110 – 3700 Gilmore Way
Burnaby, B.C., Canada V5G 4M1. (604) 430-3466

NAME__________

COMPANY__________

ADDRESS__________

CITY__________ STATE__________

ZIP__________

PCT

NCI

Portability. Reliability. Speed.

IBM and IBM XT are trademarks of International Business Machines Corporation.

UCSD p-System is a trademark of the Regents of the University of California.

the remote execution feature because there is no way to return to DOS without user intervention from within VisiCalc. In addition, any program that requires interaction with the user will not work with remote execution. However, batch processing jobs will run properly if the .BAT file is carefully constructed to return to the DOS command level upon completion. Any program that takes a file, processes it, rewrites the results to an output device, and returns to the DOS command level, should also work. This would typically include such applications as sorts, compiles, and assemblies.

PCnet Application Examples

Shared IBM PC Word Processing System This example describes a four-user word processing system that shares a hard disk and both letter and draft quality printers. In this example it is assumed that a hard disk that allows partition of its available space into multiple volumes is used. The system uses five PCs: one acts as a shared printer/mass storage workstation while the other four are used by operators to input and modify stored text. The shared PC contains a diskette drive, a fixed media hard disk unit (12 megabyte capacity) and a tape cartridge unit (5 megabyte) for backup purposes. The hard drive is formatted into 1 public volume with 2 megabytes of capacity and 4 user-owned private volumes with 2.5 megabytes each (approximately 1,600 pages of text per user). It also contains printer ports to which the two printers are attached. The volume-to-disk assignments (or mapping as the process is sometimes called) on the shared PC (sPC) is shown in table 2.

Each word processing operator's uPC is configured with only one floppy diskette drive (used primarily to boot the system). These uPCs have a volume-to-disk drive mapping as shown in table 3.

Table 2

Shared PC Disk Volume Assignments - Word Processing Example

Volume	Drive	Contents
A:	Fixed 2 Mbyte	Common programs including the word processor, mail merge, spelling checker, etc. Read only by all uPCs
B:	Fixed 2.5 Mbyte	Read/write file storage for user #1
C:	Fixed 2.5 Mbyte	Read/write file storage for user #2
D:	Fixed 2.5 Mbyte	Read/write file storage for user #3
E:	Fixed 2.5 Mbyte	Read/write file storage for user #4
F:	Cartridge 5 Mbyte	Used exclusively for backup. Can backup two users at a time
G:	Floppy	Used to boot and copy programs to shared volume A:

Table 3

User PC Disk Volume Assignments - Word Processing Example

Local Volume	Mapped Drive	Contents
A:	Network shared drive A:	Volume containing read-only programs
B:	Network shared drive X: (X=B:, C:, D:, E:)	Volume assigned as read/write storage for this user
C:	Local floppy drive	Volume for local storage

Because volume A: on the shared hard disk drive contains the common applications programs, all users will need access to that volume. Volume A: is also designated the network default volume. Typical applications programs located on Volume A: are the word processing package (WordStar for this example), mail-merge, spelling checker (Word Plus for this example), grammar checker, indexing program, and footnoter. Volume A: might also contain universal boiler plate files for insertion into documents, although each user will likely have his or her own such files as well. Each word processing operator will have a dedicated volume for the storage of required document files. For purposes of clarifying this application, the following sequence of events is presented for the user on PC#3 as a document preparation task is carried out: PC#3 operator directs printer output to the shared draft quality printer on PC#5 and requests the word processing program from the hard disk:

```
A>assign 1pt1 to PC#5 1 pt
A>ws.com
```

In response to the WordStar prompt for an edit file, the operator then requests a document file from his private volume:

```
d:chapter6.doc
```

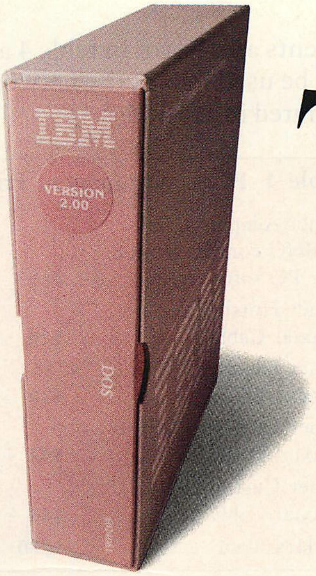
When editing is complete, the revised document is saved back to volume D:. The operator then desires to run the document through a spelling check sequence while editing on a second document is begun:

```
A>re spell d:chapter6.doc
```

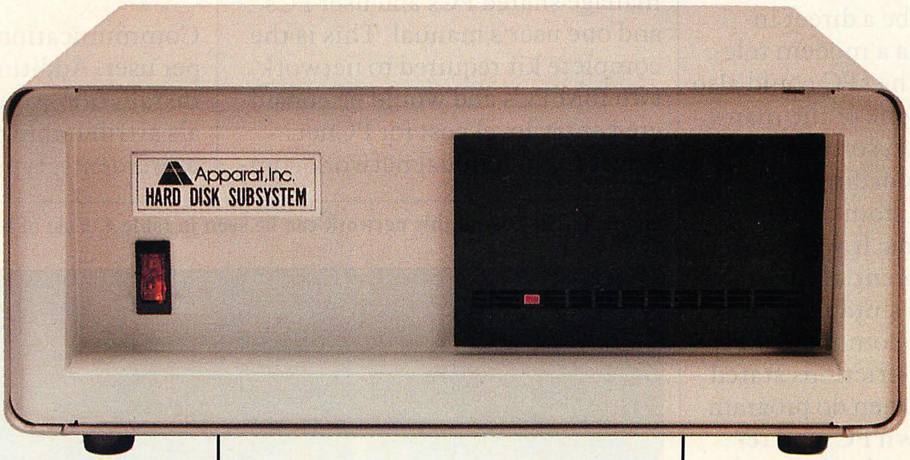
This places a file on drive D: called errwords.txt and signals the user on PC #3 when complete. After reviewing errwords.txt and inserting corrections for misspelled words, that user now has the option to issue another remote command to mark his or her file with words to be corrected:

```
A>re markfix d:chapter6.doc
```

Finally, when the operator is ready to print the draft manuscript, a WordStar print file command is issued and the desired file will print automatically on the draft quality



THIS IS ALL IT TAKES TO RUN OUR HARD DISK.



Compatibility. A lot of manufacturers talk about it. Mostly, compatibility is defined as something that almost works like the original. Our hard disk sub-system for the IBM PC runs on 2.0 without modifications, patches or use of device drivers. Just boot 2.0 and run.

And we haven't cut back on reliability either. The Apparat hard disk uses the Xebec controller and connects via a host interface module to one of the slots in your PC. The HIM also contains an RS-232 serial async port for use with a mouse, modem, etc. Even with compatibility and reliability built in, we've managed to provide some very competitive pricing. The 10 and 30 megabyte drives are priced at **$1,995** and **$3,395** respectively.

When you use a multifunction RAM Card, such as the Apparat Combo II (with up to 512K of RAM, RS232, parallel printer, clock calendar, and game adapter) along with the HIM, floppy disk controller and CRT monitor adapter cards in your PC, you still have one slot available for future expansion.

If you're considering a fixed disk for your IBM PC, look at all the systems available. Look for compatibility. Look for reliability. Look for value. We think you'll pick ours.

For more information or to order yours call **800/525-7674** or write Apparat, Inc., 4401 S. Tamarac Parkway, Denver, CO 80237, 303/741-1778.

IBM PC is a registered trademark of International Business Machine Corp.

Prices and specifications subject to change without notice.

Apparat, Inc.

CIRCLE NO. 172 ON READER SERVICE CARD

printer attached to PC#5. During this time the operator is free to continue further editing work.

Mainframe Program Development
Many large mainframe-oriented, time sharing systems are not optimized for maximum productivity in the creation and development of software. Many systems lack good, interactive full-screen editors and often provide poor response when heavily loaded. An alternative is to use IBM PCs to write programs and review results. However, mainframe communications lines are expensive and are poorly utilized if one line is provided for each PC.

A solution could be structured using PCnet to make the process more productive for a programming staff. This solution would consist of connecting a shared PC (sPC) with a hard disk and a modem to the mainframe and linking user PCs to the sPC via the network. The communications line to the mainframe could be a direct in-house access or via a modem telephone network. The sPC could also be equipped with one of the many terminal emulation software packages that allow uploading and downloading files from certain mainframe systems. It would be wise to check a potential file-transfer program for its suitability to be run by the remote command, keeping in mind the restrictions stated earlier. Users can then do program editing on their own PC, and request source code uploading and processing on the mainframe via a remote command to the shared PC. After the program is processed on the mainframe, the user requests downloading to the hard disk where it is accessible via the network for further development. In most cases this technique is limited to source code editing on the PCs; program compilation or assembly would in all likelihood be done on the mainframe itself.

The volume assignments for this example are virtually identical to those in the word processing situation. Volume A: is the network default volume and contains the required text editors, syntax pre-processors, file transfer programs and other software development aids. Each programmer has a private volume for source code and related development files. The above volumes should all be located on the shared hard disk drive.

Economics
The final topic of discussion in this review of PCnet will be the cost factors involved in the installation of the network. A starter system is available, which consists of two network adapter cards, two BNC-type "T" connectors, one finished 20-foot coaxial cable with two 75ohm terminators, one distribution diskette each to initialize and manage shared PCs and user PCs, and one user's manual. This is the complete kit required to network two IBM PCs and would be considered entry level cost for PCnet. Prices for individual network components are shown in table 4 and will be used to cost the network illustrated in figure 4.

Table 4 PCnet Component Pricing

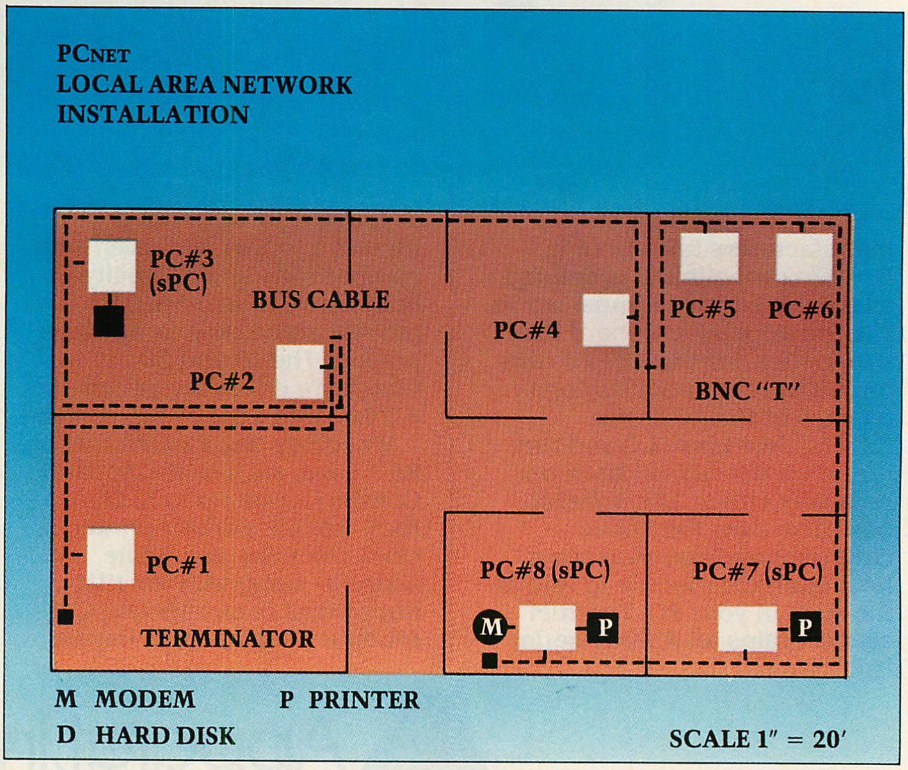
PCnet Adapter Card (w/BNC connector and user PC software)	$695
PCnet Finished 20 foot Coaxial Cable	$20
PCnet Finished 50 foot Coaxial Cable	$29
PCnet Custom RG59/U Coaxial Cable	$44 + .30/ft
PCnet Custom RG11/U Coaxial Cable	$44 + .60/ft
Tri-Hex Tool	$200

The installation in figure 4 requires the following hardware and software.

* 1 Starter Kit	$1490
* 6 PCnet Adapter Cards + BNC Hardware	$4170
* 760 feet of Customer RG59/U Coaxial Cable (measured length of bus cable plus 50% for corners, routing, taps, etc)	$ 272
Total	$5932

Communications cost is then $742 per user. Additional costs for this installation would be any peripherals attributable to networking requirements, typically a hard disk

Figure 4: The cost of this network can be seen in table 4. Cost of hardware and software is $5,932.

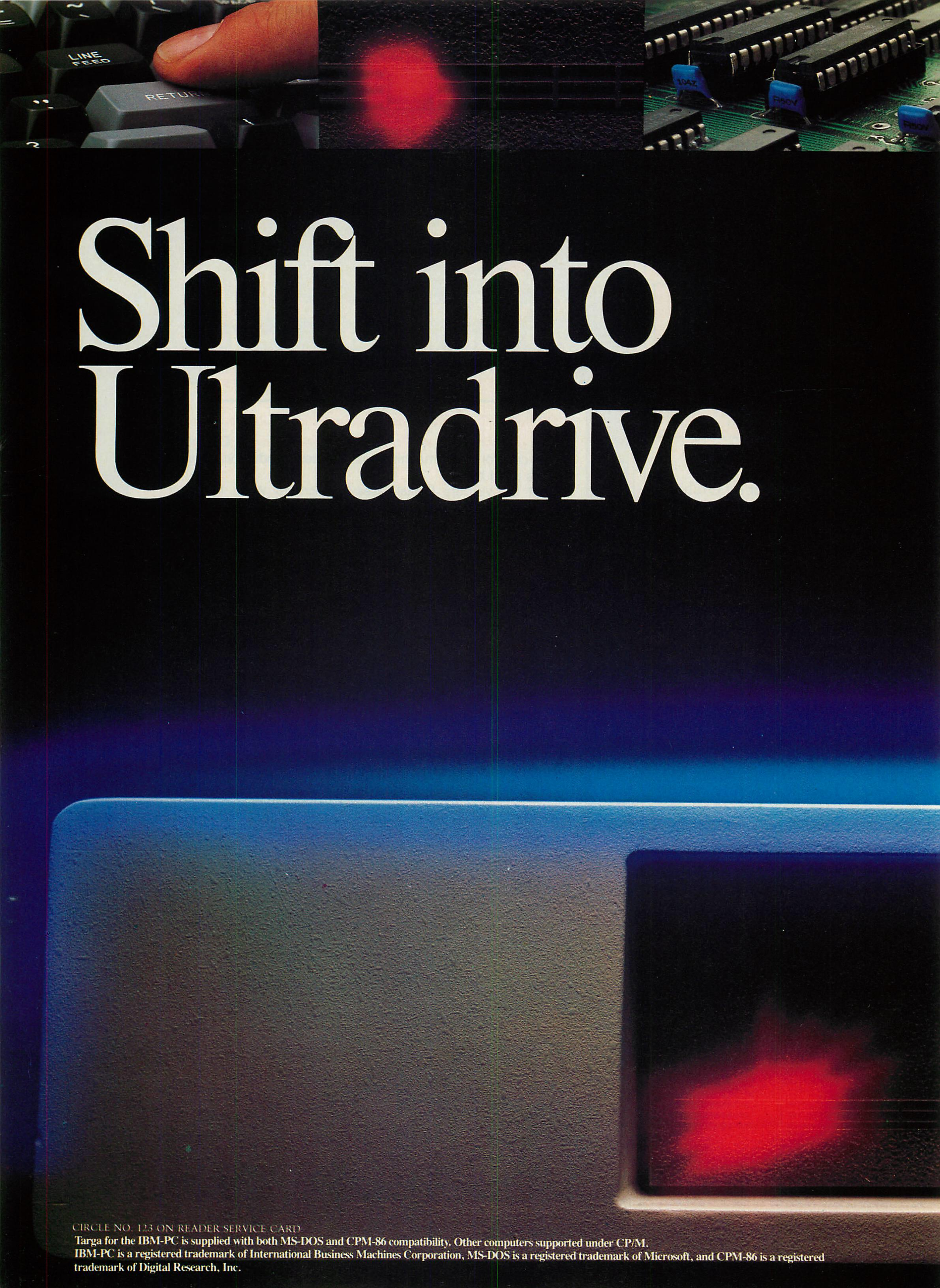


unit. Because PCnet uses any IBM PC plug-compatible peripherals, these costs will vary widely. The release of electronic mail and print spooling software will increase network costs by $750 and $600 respectively. Assuming the base case of a 760 foot pre-wired installation with starter kit at $1762, cumulative costs for adding additional capability are as follows:

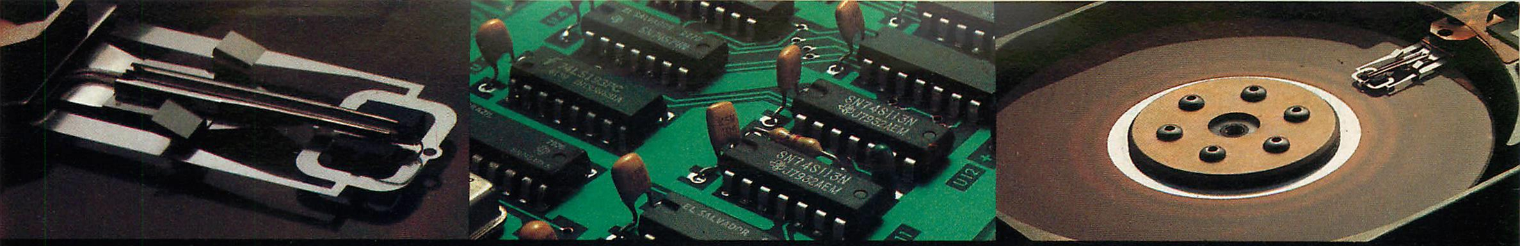
# workstations:	3	4	5	6	7
Cumulative $:	$2457	3152	$3847	4542	5237
Cost per User:	$819	788	$769	757	748

Summary

In this review of PCnet by Orchid Technology I have described how one low-cost area networking system works with the IBM PC. I have only scratched the surface of potential applications, both within the immediate network sphere and with connections to the "outside" world through a shared modem. Resource and file sharing will become increasingly important in a complex world where communications effectiveness will equate to business success. One factor missing from this review has been the discussion of applications software optimized for network operations. This was not an oversight but a reflection of an incomplete equation. As networking matures, the emphasis will shift from the technology of communications to truly integrated business software that exploits the technology. Such integration will include accounting, word processing, electronic mail, remote file transfer, and graphics. ■



LINE FEED
RETURN
Shift into Ultradrive.
CIRCLE NO. 123 ON READER SERVICE CARD
Targa for the IBM-PC is supplied with both MS-DOS and CPM-86 compatibility. Other computers supported under CP/M.
IBM-PC is a registered trademark of International Business Machines Corporation, MS-DOS is a registered trademark of Microsoft, and CPM-86 is a registered trademark of Digital Research, Inc.



Targa™: The ultimate mass storage machine for your IBM®-PC. and other microcomputers.

With up to 30 megabytes of hard disk capacity, you can handle data bases that would fill phonebooks. And access, process, and update that data in seconds.

Which, after all is what serious personal computing is all about.

The Purr of Performance.

Targa is one beautiful piece of engineering.

Fast. Data glides in and out at 5 million bits per second.

Quiet. Even with disks spinning at 3600 rpm inside.

And solid. Targa is built to deliver uncompromising performance.

Automatic Is Standard.

Just plug the Targa into your PC, and you're ready to go.

Targa comes with plain English instructions and plenty of software. Easy to use, clearly documented programs help you transfer data in and out, use powerful system utilities, and diagnose problems.

The Luxury Model at Economy Prices.

You may order your Targa directly from the CMC factory, one of the world's oldest micro-computing hard disk manufacturers, or, contact us for the name of your local dealer.

Powerful standalone Targa mass storage systems start at $1550. Considerably less than many lesser systems.

Targa 50	5 megabyte models from $1550
Targa 150	10 megabyte models from $1890
Targa 150	15 megabyte models from $2190

Prices include shipping and handling. Call for prices on expanded capacity models.

Take A Test Drive.

No risk.

We'll ship your Targa factory direct. Then give you 30 days to make sure you're satisfied. If not, we'll refund you full purchase price. Plus your return postage costs.

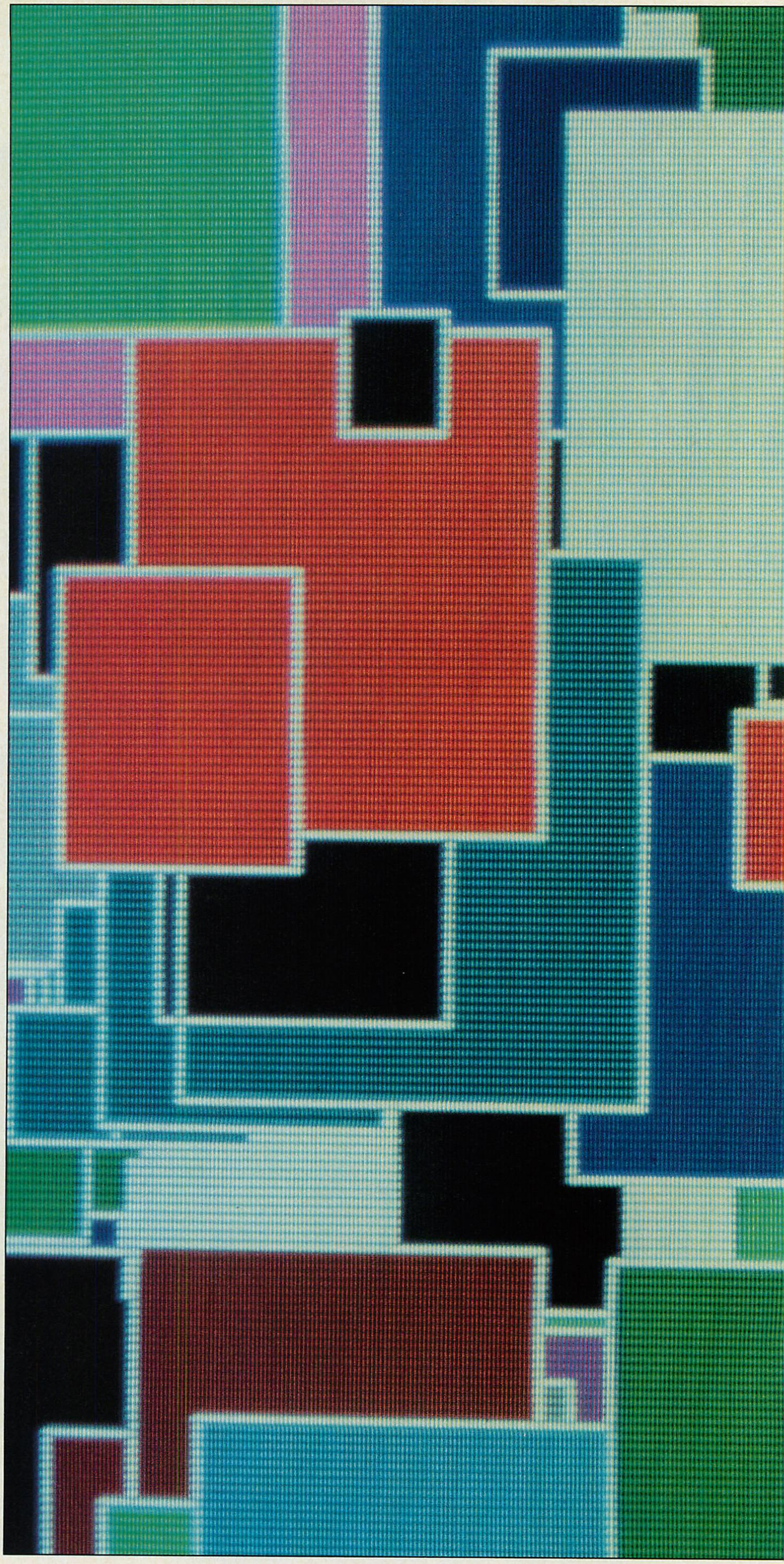
To reach our professional sales staff, call toll free, **800-CMC-INTL.** (In Washington, dial 206-885-1600.)

You can charge the Targa to any major credit card. Or you can send a check, money order, or wire transfer. Your Targa will arrive within days.

And your personal computer will get a much needed boost in mass storage horsepower.

Color, Plus

OUR RESIDENT EXPERT LOOKS AT A POPULAR ADD-IN CARD FOR THE PC



THOMAS V. HOFFMANN

BETTER READ THAN FRED

My first exposure to puns was in the first grade when our teacher, Mrs. Redling, asked us "What's black and white and red all over?"

"I don't know," I replied innocently, "what?"

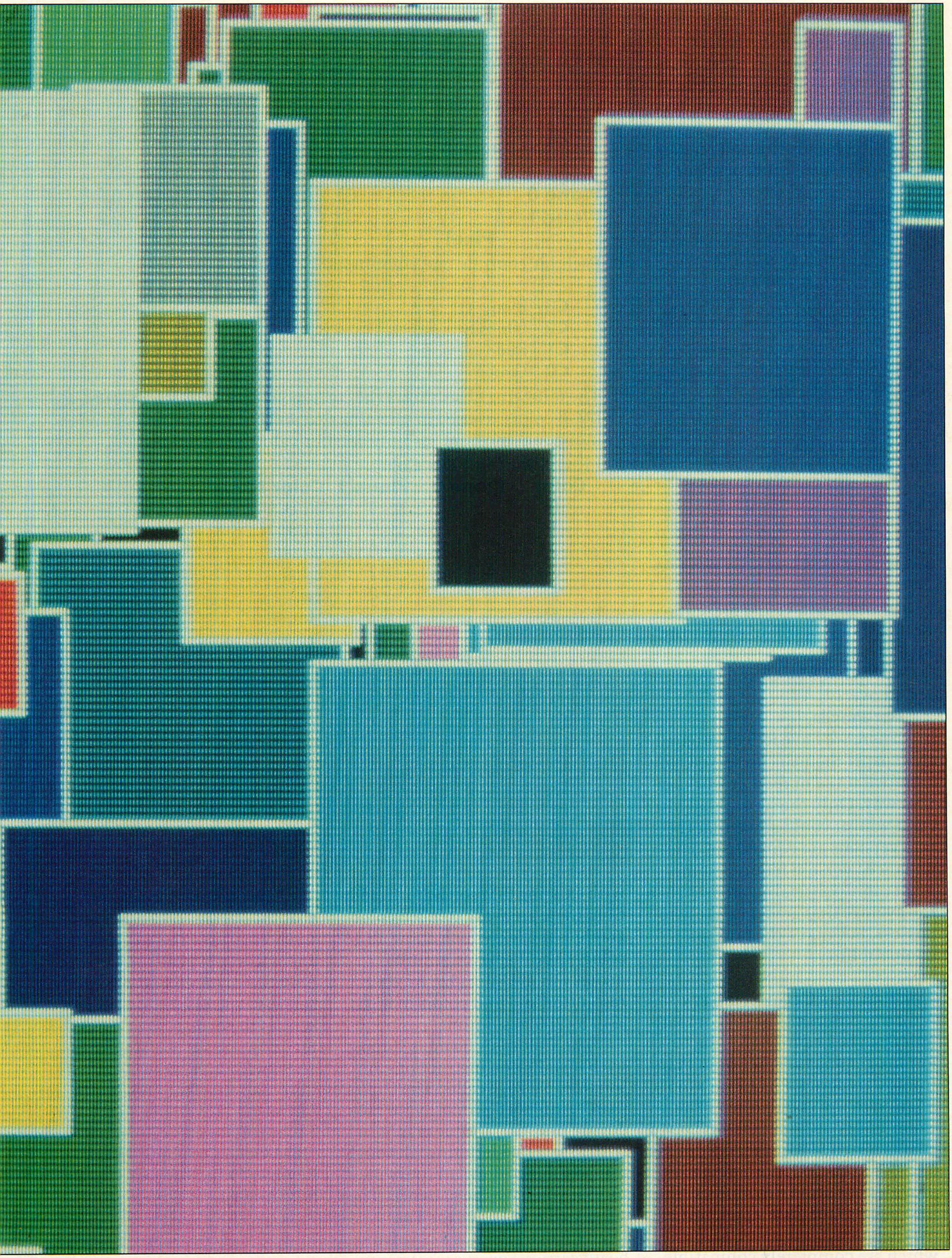
"The New York Times," she answered.

I thought about that a bit. Not much room for argument, for in those days, long before *U.S.A. Today* made the scene, newspapers were universally black and white, and the *Times* was widely read, if not by six-year-old kids from Brooklyn then surely by plenty of other people. So it was logical, if not as existentially funny as the other joke I learned from my grandfather Fred about a chicken crossing the road. Still, I felt cheated.

"But you said red," I protested.

"No I didn't, I said 'read.' " She wrote it on the blackboard. "It

Thomas Hoffmann is director of advanced systems development for General Instrument Corporation.



The image created by the program BOXES, which follows this article.

sounds the same, but it isn't spelled the same. Words like these are called homonyms. . . ,'' and so another grammar lesson began, leaving my first pun unfit to print. Until now.

A simple substitution renders the couplet media-independent,

The Colorplus board is shipped with a software package called the draftsman™, intended primarily for producing line, bar, and pie graphs with associated lettering although it is possible to draw arbitrary pictures if you have the patience.

compatible with visual presentation without danger of premature revelation:

Q: ''What's black and white and red all over?''

A: ''Pravda.''

There we have it. The original newspaper theme is preserved, you can spell it *red* or *read*, it works either way. Only the most rabid ultra right winger could argue that the two versions are equivalent. The enhanced version requires somewhat more political awareness than the average first grader is likely to possess, but that's the price of compatiblility.

The Price of Compatibility

A totally different kind of compatibility can be had for $475: the Colorplus graphics adapter from Plantronics/Frederick Electronics (no relation to my grandfather). The Colorplus board—actually it is two boards fastened together occupying one slot in the PC motherboard—provides the same functions as the IBM color graphics adapter plus a parallel printer interface, all of which are completely compatible with existing IBM software. In addition, there are two extended color graphics modes, which allow all 16 colors to be displayed simultaneously in medium (320 by 200) resolution and four colors in high (640 by 200) resolution.

The Colorplus board is shipped with installation instructions and a software package called The Draftsman™, intended primarily for producing line, bar, and pie graphs with associated lettering, although it it possible to draw arbitrary pictures if you have the patience. The suggested retail price for The Draftsman by itself is $295. The board carries a one-year warranty.

The manufacturer assumes that the average purchaser is not interested in how the board works, and will simply use the software provided or purchased separately. A software specification that describes the operation of the extended modes is available on request without charge. Detailed information on the standard operating modes is available in the IBM Technical Reference Manual and in ''IBM Color Graphics: an In-Depth Review,'' which appeared in the July/August *PC Tech Journal*.

Table 1.
Colorplus Medium Resolution Extended Graphics

Corresponding bytes from each plane each contribute two bits per pixel. As in the standard IBM medium resolution mode, the high order bits determine the left-most displayed pixel.

	Bit	7	6	5	4	3	2	1	0
Plane 0		C1	C0	C1	C0	C1	C0	C1	C0
Plane 1		C3	C2	C3	C2	C3	C2	C3	C2
	Pixel	0		1		2		3	

Each bit controls one color input as follows:

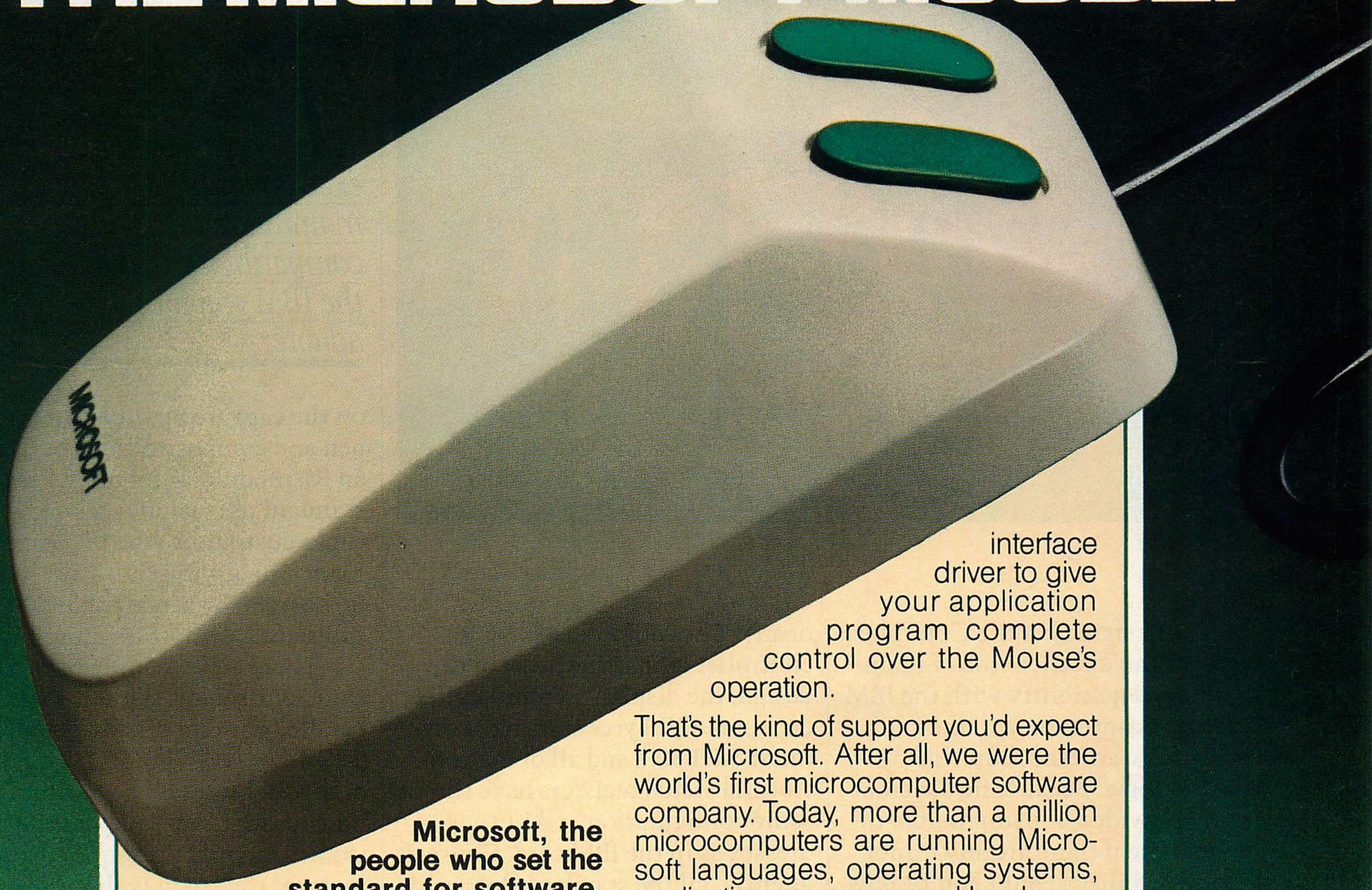
C3 Blue
C2 Intensity
C1 Red
C0 Green

The following chart shows the correspondence between the standard IRGB colors (as used in selecting text and background colors) and the individual medium resolution color values for each plane.

I-R-G-B Medium Res Color

Color	Plane 0	Plane 1	Color Name
0	0	0	Black
1	0	2	Blue
2	1	0	Green
3	1	2	Cyan
4	2	0	Red
5	2	2	Magenta
6	3	0	Brown
7	3	2	White (Light Gray)
8	0	1	Dark Gray
9	0	3	Light Blue
10	1	1	Light Green
11	1	3	Light Cyan
12	2	1	Light Red
13	2	3	Light Magenta
14	3	1	Yellow
15	3	3	Intense White

THE NEW STANDARD.
THE MICROSOFT MOUSE.



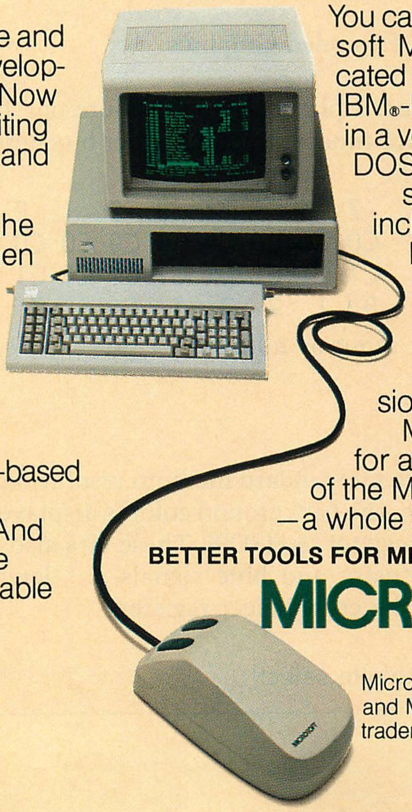
Microsoft, the people who set the standard for software, have done it again with the Microsoft® Mouse.

Our expertise in both hardware and software has gone into the development of the Microsoft Mouse. Now you can plug in the most exciting computer product of the year and put it to work.

The Mouse lets you move the cursor freely and naturally, then execute commands at the push of a button.

The Microsoft Mouse is a complete system. It comes with an on-screen tutorial, a practice application, and the Multi-Tool™ Notepad, a mouse-based text editor, so you can begin using the Mouse right away. And for application developers, the Mouse includes a programmable interface driver to give your application program complete control over the Mouse's operation.

That's the kind of support you'd expect from Microsoft. After all, we were the world's first microcomputer software company. Today, more than a million microcomputers are running Microsoft languages, operating systems, application programs, and hardware-software combinations.

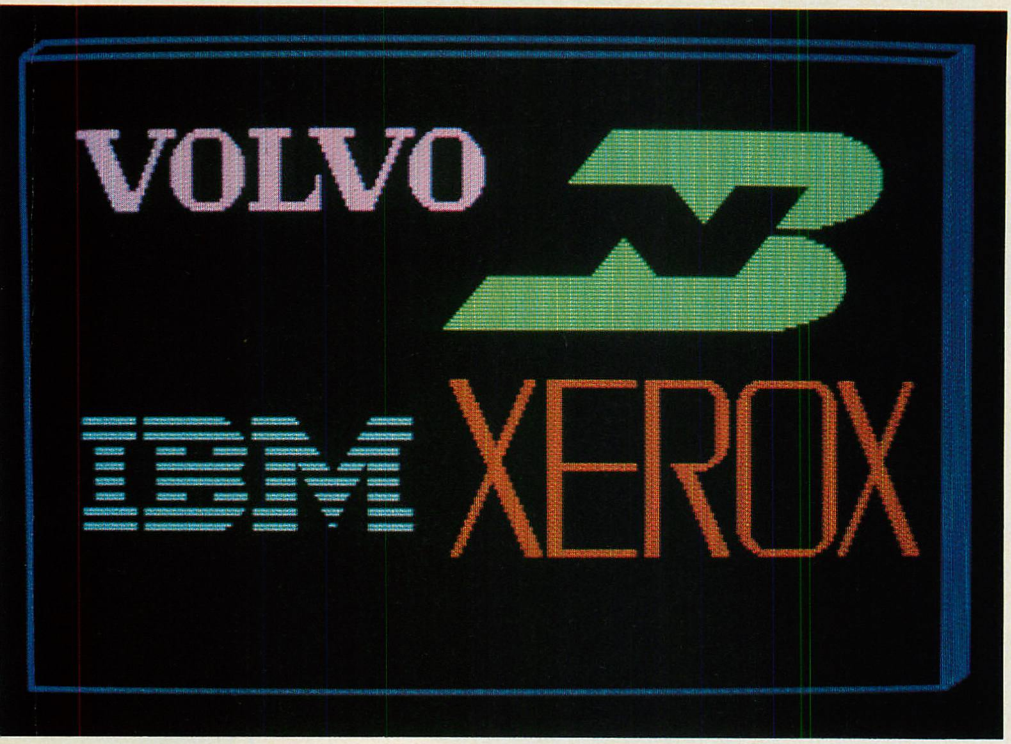


You can get the Microsoft Mouse in dedicated versions for the IBM®-PC, PC XT, and in a version for MS™-DOS machines with serial interfaces, including the IBM-PC. The Mouse supports all versions of MS-DOS, including version 2.0. Ask your Microsoft dealer for a demonstration of the Microsoft Mouse —a whole new standard.

BETTER TOOLS FOR MICROCOMPUTERS

MICROSOFT™

Microsoft is a registered trademark, and MS and the Microsoft logo are trademarks of Microsoft Corporation.



The designers of the Colorplus board did an excellent job of maintaining compatibility with the IBM graphics adapter. In fact, the card seems to be an exact duplicate of the IBM logic. Because no schematics are provided, it's impossible to tell for sure if there are any minor incompatibilities, but I don't know of any. The Motorola 6845 CRT controller chip is the heart of the design, the display memory buffer occupies 32K bytes starting at segment &HB800, and all of the control and status registers have the same configuration and I/O addresses as in the IBM card. Consequently, all the standard alpha and graphics modes function exactly as they do in the IBM card.

The rear panel of the card contains a 9-pin connector for an RGB monitor and a 25-pin connector for the printer interface. Connectors

The designers of the Colorplus board did an excellent job of maintaining compatibility with the IBM graphics adapter.

on the card are provided for a light pen and composite video output to an RF modulator for use with a standard television receiver. To those customers who request one, Plantronics will send—currently at no charge—a special cable that will interface a composite monitor.

There are two jumpers on the card. One selects the address of the parallel printer port and &H3BC (LPT1:) or &H378 (LPT2:), but none of the literature tells where it is or how to set it. A phone call to Plantronics elicited the required information: the board is shipped as LPT2. If it is to be your only printer interface, it will be more convenient to switch the jumper to LPT1.

The other jumper selects either a single- or double-dot character font. Figure 1 illustrates the difference between the two fonts. The double dot font, used on the IBM adapter, is constructed so that every horizontal segment of the character is at least two pixels wide. This ensures better color rendition on low resolution composite displays. The single dot font looks sharper on high resolution RGB displays, but exceeds the video bandwidth of low resolution color displays, resulting in messy characters with fringes of false artifact colors. The board is shipped with the single dot font selected, on the assumption that people serious

Table 2.
Colorplus High Resolution Extended Graphics

Corresponding bytes from each plane each contribute one bits per pixel. As in all other graphics modes, the high order bits determine the left-most displayed pixel.

Bit	7	6	5	4	3	2	1	0
Plane 0	C0	C0	C0	C0	C0	C0	C0	C0
Plane 1	C1	C1	C1	C1	C1	C1	C1	C1
Pixel	0	1	2	3	4	5	6	7

Each bit controls one color input as follows:

C1 Red
C0 Green

Colors are formed as they are for standard medium resolution mode. When C1 and C0 are both '0', the background color is displayed from bits 0-3 of the color select register (&H3D9). These bits also set the border area color. The intensity and blue signals are determined globally by bits 4 and 5 of the color select register.

Intensity CSR bit 4
Blue CSR bit 5 (palette select)

enough about graphics to buy the Colorplus board will also have invested in a high resolution monitor. It may not be a valid assumption, but it's pretty good advice.

Extended Color Modes

The extended color modes, especially the medium resolution 16-color mode, are what make the zontal resolution. I don't think the extra programming effort required is worth the trouble for most situations. The exception would be where there is a requirement for 80-character lines of text combined with graphics on the same screen.

The Planes, The Planes

The key to the extended color

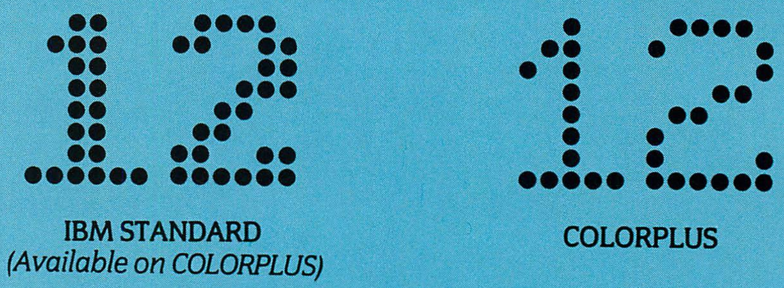


Figure 1: Single- and double-dot character fonts. The double dot font used on the IBM adapter ensures better color rendition on low resolution composite displays.

board worthwhile. The IBM scheme, which allows only one of two predetermined color sets plus any one of the 16 background colors, makes the generation of pleasing images for games, education, and business presentation graphics a much greater challenge than it ought to be. Having the full color set available removes a heavy burden from the image designer.

The board is shipped with the single-dot font on the assumption that people serious enough about graphics to buy the Colorplus board will also have invested in a high resolution monitor.

The extended high resolution mode provides the same color choices as the IBM medium resolution mode but with twice the hori-

modes is the 32K display buffer, twice as big as IBM's. The extra memory doubles the number of bits per pixel, thus four bits give 16 colors in medium resolution, and two bits give four colors in high resolution.

In the extended modes, the display buffer is treated as two identical, overlapping planes, each contributing half of the bits for each pixel. The display electronics accesses both planes in parallel, taking corresponding pixels from each and combining them to form the displayed image.

The pixel arrangement in each plane in extended color modes is the same as for the corresponding standard mode. This makes it convenient to use BASIC graphics statements to draw images or text. But BASIC only knows about a 16K display buffer, so how can it access the second plane?

The extended mode/plane select register, at I/O address &H3DD, contains three active bits: one for each of the two extended modes, and another for controlling the addressing of the extended buffer memory.

METAFILE® takes your PC *beyond computing.*

METAFILE is the high-level language that provides all of the software facilities a PC owner requires in a *single integrated architecture.* METAFILE enables you to go. . .

BEYOND PACKAGES — Standard facilities include:
- Data base management
- Word processing
- Impromptu inquiry
- Forms data entry
- Sorting
- Report generation
- Mailing lists • Spread sheets
- Network communications

BEYOND WORD PROCESSING — Any text can be a:
- Document • Menu
- Form • Procedure • Report
- Active text with embedded descriptions

BEYOND PROGRAMMING — Develop applications with:
- Easy path from prototype to production
- Documentation produced within the same system
- Immediate turnaround
- Built-in tracing and troubleshooting aids
- Visual prompting

BEYOND DATA BASE MANAGEMENT — Standard facilities include:
- Virtual updates
- Data conversion to foreign formats
- Data base reorganization
- Query by example
- View management

For more information contact:
Chicago 312/724-0310; Philadelphia 215/644-4420;
Minneapolis 612/854-2272; Rochester, MN 507/281-3191;
New York City 212/674-1341; Dallas 214/373-0533;
Columbus 614/766-1101

CIRCLE NO. 168 ON READER SERVICE CARD

METAFILE®

Beyond Computing™

METAFILE®
Olmsted Federal Building
Chatfield, MN, USA 55923
Telephone: 507/867-4440

Bit 6 Selects the active display memory plane. When it is '0' the first 16K (plane 0) is at address &H0000 and the second 16K (plane 1) follows it at &H4000. A '1' switches the two planes, so that plane 1 is at address &H0000, and plane 0 at &H4000.

Bit 5 Selects extended high resolution (640 by 200) mode when set to '1'.

Bit 4 Selects extended medium resolution (320 by 200) mode when set to '1'.

When bit 4 and 5 are both '0', the adapter operates in normal IBM-compatible mode. Bits 4 and 5 should never both be '1'. Bits 7 and 0-3 have no effect and should always be '0'. The extended mode register is reset automatically whenever any alpha mode is selected—SCREEN 0 in BASIC, or bit 1 of the standard mode register &H3D8 set to '0'.

Note that all 32K bytes of buffer memory are accessible to the CPU at any time. It is not necessary to switch planes when using PEEK and POKE to manipulate the memory. Similarly, an entire image of 32,768 bytes can be BSAVEd or BLOADed in one piece. This makes managing a disk full of image files a lot simpler. (At least one supplier of software for the Colorplus board doesn't seem to know about this.) Plane switching is required only to coerce BASIC—or other languages with standard IBM assumptions built in—into accessing the second plane.

The sequence of instructions for establishing either of the extended modes is as follows:

1. Set the corresponding standard graphics mode (medium or high resolution). In BASIC this is done by using SCREEN 1 or SCREEN 2, respectively.
2. Set the extended mode/plane select register to select the desired extended mode.
3. Set the color select register (&H3D9) to the desired value. In extended medium resolution mode, bits 0-3 select the border color. In extended high resolution mode, bit 5 selects the foreground color set ('0' for green, red, brown; '1' for cyan, magenta, white), and bits 0-3 select the background and border color.

Tables 1 and 2 show the contribution of bits in each picture plane to the corresponding pixel color for the extended graphics modes.

The program BOXES displays randomly placed squares, painted with successive colors, in medium resolution 16-color mode. The strange logic involving C, C0, and C1 maps the standard IBM color number (C) into the color values for planes 0 and 1 (C0 and C1, respectively).

The Colorplus seems well-designed and well-constructed, and is reasonably priced, but the one-year warranty—four times the IBM term—may unfortunately be the most valuable feature.

Caveat Emptor

The Colorplus seems well-designed and well-constructed and is reasonably priced, but the one-year warranty—four times the IBM term—may unfortunately be the most valuable feature. Perhaps its just my luck, but the first board I received was incapable of displaying a coherent image, though the printer interface worked fine.

A call to Plantronics revealed that I had one of the first 50 pre-production prototypes, and a change had been made (to make it work?). They promptly sent another, which seemed to work fine, until I tried it with a television and RF modula-

tor. The image was very unstable and filled with a regular, snowy interference pattern.

Another phone call. Does the board work with TVs and RF modulators? Well, they thought it should, but hadn't ever actually *seen* one connected to a TV. I was told to try another TV; their Amdek composite monitor worked alright. It looked worse on my other TV. The picture on the IBM Color Display looks fine.

> *A word to the wise: If at all possible, try the board you intend to buy with the equipment you intend to use—before you take it home.*

Meanwhile, I've noticed one more problem: some bits in the graphic memory seem to change spontaneously to '0'. This shows up as little black dots in what should be a solid area of color. Reloading an image file usually restores the bits to their proper configuration. This is much more than a minor annoyance; consider PAINTing a supposedly enclosed area when one of the boundary bits suddenly disappears, and the paint escapes, uniformly coloring hours of work.

Now, I don't think all Colorplus boards have these problems, but I doubt mine is unique. There are plenty of locical explanations for occasional lapses in quality—even IBM has them—and this is certainly not a valid statistical sample. The people at Plantronics are helpful and responsive, so I'm sure things will eventually get straightened out. But a word to the wise: if at all possible, try the board you intend to buy with the equipment you intend to use—before you take it home. And, of course, know your dealer.

Plus Software

The following software products were designed for use with the Colorplus board.

Business Graphics

The Draftman™ for Colorplus™
- Menu-driven
- Accesses VISICALC (DIF) data files
- HP-7470A Plotter output
- IBM/EPSON Printer output
- Operator's manual on diskette
- Sequential display of saved screen images (SOFTWARE SLIDE SHOW)
- Power EDIT function for custom graphs and image creation

Starware & PC+ Products—$295

CIRCLE NO. 376 ON READER SERVICE CARD

Business Graphics System™
- Menu-driven
- Accesses Peachtree™ and Supercalc™ data files
- Multiple character fonts
- 3-d bar charts
- Multiple plot capability
- Supports numerous plotters and Epson printers

Peachtree Software Inc.—$295

CIRCLE NO. 375 ON READER SERVICE CARD

(Continued on page 78)

TECH JOURNAL PROGRAM LISTING #1

```
100 'File: BOXES2.BAS
110 'Auth: T. Hoffmann -- 13 June 1983
120 '
130 ' Illustrates COLORPLUS extended medium resolution
140 ' graphics mode (300 by 200 by 4, 16 colors)
150 '
160 '
170 '--- Definitions
180 '
185 DEFINT A-Z
190 COLORREG = &H3D9     '-- Color Select Register
200 EMODEREG = &H3DD     '-- Extended Mode Register
210 MRES = &H10          '-- Medium Res
220 HRES = &H20          '-- High Res
230 PLANE1=&H40          '-- Plane select
235   P0=MRES: P1=MRES+PLANE1
240 '
250 '-- Select Color Adapter
260 '   and Extended Medium Resolution
270 '
275 MONO = 1    '-- Set to 0 if no monochrome adapter
280 GOSUB 9000
290 SCREEN 0: SCREEN 1
295 KEY OFF
300 OUT EMODEREG, P0: CLS: OUT EMODEREG, P1: CLS
310 OUT COLORREG, 0
320 '
330 '-- Repeat Until F10 Key
340 '
350 KEY (10) ON: ON KEY (10) GOSUB 9900
360 WHILE 1
370   '-- Generate Position, Size, and Color
380   X=RND*300: Y=RND*200
390   R=60*RND+10
400   C=(C+1) MOD 16
410 '
420   '-- Separate Colors for Each Plane
430   C0 = (C AND 6) / 2
440   C1 = (C AND 8) / 8 + (C AND 1) * 2
450 '
460   '-- Draw and Paint in Each Plane
470   OUT EMODEREG, P0
480   GOSUB 600: PAINT (X+3,Y+3),C0,3
490   OUT EMODEREG, P1
500   GOSUB 600: PAINT (X+3,Y+3),C1,3
510 '
520 WEND
600 '-- Draw Box with color 3, inside color 0
610   LINE (X,Y)-STEP (R,R),3,BF
620   LINE (X+1,Y+1)-STEP (R-2,R-2),0,BF
630 RETURN
9000 '----------
9010 ' Switch to Color/Graphics Display
9020 DEF SEG=0: A=PEEK(&H410): POKE &H410,(A AND &HCF) OR &H20
9030 WIDTH 40: SCREEN 1: SCREEN 0: LOCATE ,,1,6,7
9040 RETURN
9050 '----------
9060 ' Conditionally Switch to Monochrome Display
9070 IF MONO <> 1 THEN LOCATE 1,1: RETURN
9080 DEF SEG=0: A=PEEK(&H410): POKE &H410,(A OR &H30)
9090 WIDTH 80: LOCATE ,,1,12,13: SCREEN 0,0,0
9100 KEY ON
9110 RETURN
9900 '----------
9910 ' F10 Gets Here to Exit
9920 '
9930 GOSUB 9050 '-- return to monochrome
9940 END        '-- and quit
```

(Continued from page 76)

Hypergraphics + ™
Tool for presentations, tutorials and computer-aided instruction
175 pages of color graphics
Sophisticated and easy-to-use animation
Automatic execution of other software
DLW Corp. (call for price)
CIRCLE NO. 374 ON READER SERVICE CARD

Graphpower™
Menu-driven
HP-7470A Plotter output
Accesses VISICALC (DIF) DSS/F and PASCAL data files
Sequential display of saved screen images (SOFTWARE SLIDE SHOW)
Multiple plot capability
Variable text size & slants
Ferox Microsystems—$295
CIRCLE NO. 373 ON READER SERVICE CARD

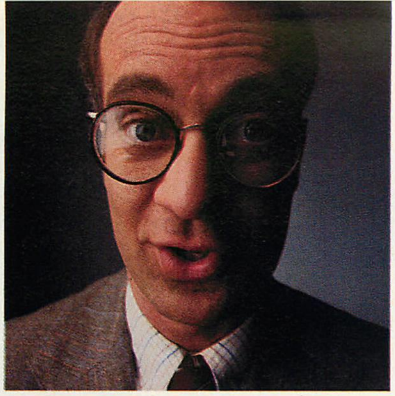
Videogram™
Design presentation quality screen images and business graphics in 16 colors
Interchangeable character fonts
Magnification of graphics and text
Image manipulation and fine painting
Softel, Inc.—$300
CIRCLE NO. 372 ON READER SERVICE CARD

Graftalk™
Menu-driven or interactive
User-modifiable menus
Multiple plotter/printer support
Integrated mini spreadsheet and editor
Slide show capability
Interfaces to ASCII data files e.g., SUPERCALC files
Redding Group, Inc.—$450
CIRCLE NO. 371 ON READER SERVICE CARD

Presentation Business Graphics System (PBG System)™
Menu-driven
Numerous display options for each graph
Choice of 2-d or 3-d display
Automatically created master directory of saved graphs
Slide show capability
Supports numerous plotters
Uses data files from D-BaseII™, SuperCalc™, and other popular spreadsheet programs
GANESA Group International—$1,500
CIRCLE NO. 370 ON READER SERVICE CARD

OPERATING SYSTEMS
UCSD P-System™
Colorplus Driver
Turtle graphics
Tektronix plot 10 emulation
Network Consulting Inc.—$100
CIRCLE NO. 369 ON READER SERVICE CARD

(Continued on page 81)



"GET IBM 3270, 3780 AND HASP COMMUNICATIONS—ADDING A ONE-BOARD PROCESSOR TO MY PC?"

"YOU CAN WITH PERSYST."

The Persyst DCP/88. The only compatible front end communications processor that can support a range of IBM communications.

It's a Persyst exclusive. The DCP/88 distributed communications processor. A one-board computer that can be configured by software to handle communications between a PC and another computer system.

In fact, the DCP/88 enables the IBM or TI PC to support a range of sophisticated IBM communications—all with just one processor.

So it gives corporate users unprecedented, low-cost flexibility to incorporate the PC into virtually any environment.

Convert any PC into an IBM 3270 terminal. An IBM 2780/3780 RJE workstation. Or a HASP/RJE workstation—instantly.

Just combine the DCP/88 with Persyst software—and you can connect your PC to any compatible host mainframe or minicomputer.

PC/3270 enables your PC to emulate an IBM 3274 Cluster control unit and 3278 terminal, supporting up to four additional devices. Including a printer and three other PCs functioning as 3278 terminals.

PC/3780 converts your PC into a 2780/3780 remote job entry terminal that can send and receive batch files to and from the host.

And with PC/HASP, your PC instantly becomes a full-function HASP/RJE workstation that can support up to seven input and seven output multi-leaved job streams concurrently.

Capability to support your communications needs now. And in the future.

Because the DCP/88 supports bisync, SDLC, HDLC and async protocols, it is the only communications processor you'll ever have to buy.

Use it to connect your PC to an expanding network of IBM communications. Configure your system to meet virtually any communications need—without adding another piece of hardware. Or expense.

All of which makes the DCP/88 as practical as it is powerful.

The Persyst Coax/3278. Still another way to expand your IBM communications.

Here's another fast way to incorporate the PC into existing SNA or bisync data processing installations.

The Coax/3278. A single-slot expansion board that converts your PC into an IBM 3278 display terminal. And with its high-speed coaxial port, you can connect the PC directly to IBM 3274 or 3276 cluster controllers.

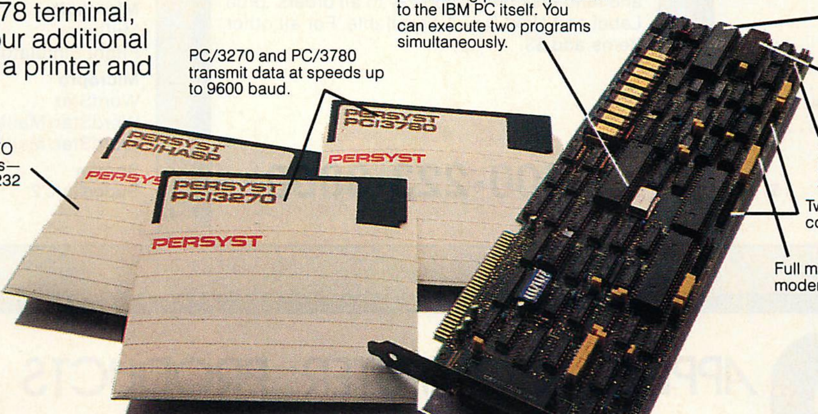
Sophisticated communications products for the corporate user. Insist on Persyst.

Persyst communications products are designed to make corporate information systems more powerful. And more efficient.

Write or call Persyst today for complete information.

PERSYST™

Persyst Products, Personal Systems Technology, Inc., 15801 Rockfield Blvd., Suite A, Irvine, CA 92714. Telephone (714) 859-8871. Telex: 467864.



THE DCP/88™ BY PERSYST.

CIRCLE NO. 184 ON READER SERVICE CARD

(Continued from page 78)

GSX Extension to CPM-86™
Implementation of GKS graphics standard
Digital Research—$60
CIRCLE NO. 368 ON READER SERVICE CARD

CLICS Colorplus Driver
C language implementation of the Core System graphics standard
Microcraft System, Inc.
CIRCLE NO. 367 ON READER SERVICE CARD

Peachtree Graphics Lanaguage™
Powerful programming language for custom graphics packages
Shares features of Business Graphics System
Peachtree Software Inc.—$395
CIRCLE NO. 366 ON READER SERVICE CARD

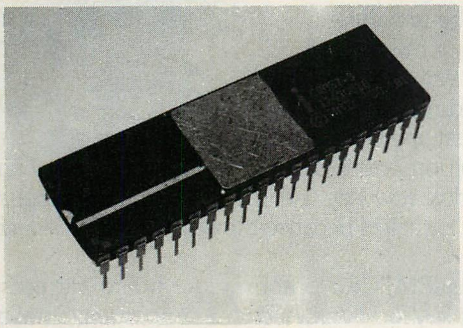
OTHER SOFTWARE

Color-It™
Screen dump to IDS Prism™ color printer
M.A.P. Systems—$49
CIRCLE NO. 365 ON READER SERVICE CARD

Teligraph™
Telidon-NAPLPS videotext standard
Graphics creation and transmission with on-line decoding
Epson printer output
Redefinable character sets
Microtaure—$398.95
CIRCLE NO. 364 ON READER SERVICE CARD

Stockview™
Plots one week's performance to 6 financial indices
Softel, Inc.—$29
CIRCLE NO. 363 ON READER SERVICE CARD

Statmap™
Menu-driven statistical mapping system
Matches geographic data bases e.g., SMSA with coordinate files
User definable range values
Zoom capability for blowing up map segments
Multiple plotter support
Ganesa Group International—$2,500
CIRCLE NO. 362 ON READER SERVICE CARD



Accessing DOS from Pascal

TECH NOTEBOOK 3

Pascal programmers can access DOS directly, but not without knowing something about the internals.

RICHARD A. LEVARO

Basic Disk Operating System (DOS) functions perform a wide variety of tasks including keyboard and screen input and output, file handling, and some special utilities. For example, there are functions to read or to set the time and date and to send output to the standard output device or printer. With more effort, file handling also can be done. DOS 2.0 has even more functions available than earlier DOS versions, and many of them are easier to use. Appendix D in the DOS Manual describes each function and how to access it.

DOS functions are accessed by placing the function number in the AH register and any other needed information in the other general registers and then executing a software interrupt (Hex 21). Information is returned in the standard registers. The DOS Manual describes the use of the registers for each function, but the documentation is geared to the assembly language programmer. What about Pascal programmers?

IBM Pascal provides an external function, DOSXQQ, which allows the user to call DOS functions from Pascal programs. To say that the IBM Pascal description of this function is brief is at best an understatement. The function is described on page 11-23 of the Pascal Manual:

```
FUNCTION DOSXQQ
  (COMMAND:BYTE;PARM:
  WORD):BYTE;
```

This function can be used to invoke the IBM Personal Computer DOS directly.

That's it. Absolutely no indication of how this calling sequence relates to the calling sequences described in the DOS manual is provided. The answer is found by studying the file ENTX6S.ASM; it is on the PAS1 diskette that is part of the compiler package. ENTX is the Pascal runtime system control, and it is written in Macro Assember. It does much of the housekeeping that's required before the Pascal program takes control. A number of PUBLIC variables are declared and initialized, and the DOS function call is constructed. DOSXQQ is the procedure listed toward the end of the file.

Looking carefully at DOSXQQ reveals that the parameter COMMAND corresponds to the AH register, and PARM corresponds to the DX register. The byte returned is the value of the AL register after the call. For example, to call function 8 (Keyboard input), the Pascal call is:

```
al := dosxqq(#8,0);
```

where al is a byte variable. To display a string, the following code works:

```
al := dosxqq(#9, adr 'Hello$');
```

The prefix adr returns the address (offset) of the string, which is what DOS expects in the DX register.

Using this information, many DOS functions can be used. However, some of the DOS functions use the CX and DX registers to return information as well, but nowhere is that provided in the Pascal calling sequence. Further investigation of the file ENTX6S.ASM shows that the CX register corresponds to the external (global to everything) variable CRCXQQ and the DX register corresponds to CRDXQQ. If the function call returns information in the CX register, it can be found in CRCXQQ. The date and time utilities (function calls 2A through 2D) are good examples. For example, to get the time,

```
al := dosxqq(#2C,0);
```

does the job. Where is the information hours, minutes, and seconds? Declaring CRCXQQ and CRDXQQ to be external word variables, they can be inspected to get the information. The DOS Manual describes the format in which the information is stored.

In summary, Pascal does provide a mechanism to access DOS directly, but the user must know something about the internals to make full use of it. IBM deserves brickbats for its documentation. Unfortunately, the story is even worse for DOS 2.0; it provides many more functions, some of which use the BX register.

The Pascal function makes no provision for the BX register (and others, like SI and DI), so many of the useful function cannot be accessed. The solution is to write a general DOS interface routine in assembler, then do all the DOS calls from Pascal to the assembly language procedure. ■

Richard A. Levaro works for Blaise Computing Inc. in Berkeley, California. The procedure he describes for accessing DOS is used by Blaise in its Pascal Tools packages.

SYSTEMATIC PROGRAMMING IN EDISON

PER BRINCH HANSEN

The Edison-PC System is a portable software system for the IBM Personal Computer and the Compaq Portable Computer. The software is written in the programming language Edison, which is simpler but more powerful than Pascal.

The Edison System is similar in capability to DOS 1.1, CP/M, or UCSD Pascal. It uses two diskettes to hold compiled programs and text files.

THE EDISON-PC SYSTEM

The Edison-PC System consists of an operating system, an Edison compiler, a screen editor, a text formatter, a print program, and an 8088 assembler. With the exception of a small system kernel, all the software is written in the programming language Edison.

The Edison System is similar in capability to DOS 1.1., CP/M, or UCSD Pascal. It uses two diskettes to hold compiled programs and text files. Because most of the system programs fit on two disks, it is possible to compile, edit, format, and print text without changing the diskettes in the drives. The sectors of each file are chained together in tables stored on the diskettes. Consequently, it is unnecessary to move files to find room for new files. The sectors of a file are separated by physical gaps of three sectors to reduce rotational delays during program loading and data transfers.

An Edison program can call other Edison programs stored on the disk. Programs are loaded into memory when they are called and are removed again after their execution. This facility makes it possible to write operating systems and multipass compilers in Edison.

The system runs on an IBM Personal Computer with 64K bytes, monochrome display, dual disk drives, and printer. On this configuration, the system can edit and compile itself, and it can also be used to develop other small operating systems.

The Edison compiler generates portable object code, which is interpreted by a kernel written in assembly language. The kernel can be edited and assembled on the system. The software was originally written for the LSI 11 Computer with 8″ floppies and was moved to the IBM-PC with 5-¼″ floppies in less than a month.

The source text of all the software is available for programmers who need to modify and replace parts (or even all) of the software. The software is simple enough to be studied at all levels of programming and is described in detail in my new book entitled *Programming a Personal Computer* (Prentice-Hall Inc.).

THE EDISON LANGUAGE

The programming language Edison was developed from Pascal but is much simpler than Pascal. Edison includes the following well-known programming concepts:

- **Standard types (integer, boolean, char)**
- **Type definitions (enumerated types, arrays, records, sets)**
- **Constants, variables, expressions**
- **Empty statements, assignments, procedure calls**
- **If statements, while statements**
- **Procedures, functions**

The programming language Edison was developed from Pascal but is much simpler than Pascal.

To make the language as simple as possible the following Pascal-features were excluded from Edison:

Reals, subrange types, pointer types
Variant records, files
Goto, case, repeat, for, with statements

In spite of these omissions, Edison is more powerful than Pascal due to the following new features:

Concurrent statements
When statements
Modules
Library programs
External procedures

This article will concentrate on the features that make it possible to write operating systems and concurrent programs in Edison.

Disk Files

In Pascal, file operations are built into the language as standard procedures (read, write, get, put). Because Edison is used to write operating systems, it must be possible to program the input/output procedures in the language itself. Consequently, files are not built into the language.

The Edison-PC System defines a data type named stream, which other programs use to access text files stored on diskettes. A program declares a file as a variable of type stream

```
var file: stream
```

and calls the following operating system procedures to input the file:

openread(file, title)	Opens a file with a given title as an input stream.
more(file)	Has the value true, if there are more characters in an input stream; otherwise the value is false.
read(file, value)	Reads the next character in an input stream.

The operating system implements similar procedures for output streams. Notice that you can write other operating systems in Edison that implement completely different input/output procedures.

Concurrency

Let's look at an example to see how concurrency and modularity work in Edison. We'll write a small program with two processes that cooperate on the same task: one process inputs a text file from a diskette and sends it through a message buffer to the other process, which displays the text on the screen.

A procedure describes the input process:

```
proc input(title: name)
var file: stream; value: char
begin
  openread(file, title);
  while more(file) do
    read(file, value);
    send(value)
  end;
  send(etx)
end
```

The parameter called title defines the name of the file. A local variable named file describes the file as a stream. The procedure inputs one character at a time and sends it through a buffer. The last character sent through the buffer is the control character etx (end of text). We will look at the send procedure later.

The procedure that describes the output process looks like this:

```
proc output
var value: char
begin
  receive(value);
  while value <> etx do
    display(value);
    receive(value)
  end
end
```

It receives one character at a time from the buffer and calls an operating system procedure to display it.

When you start the execution of an Edison program, it begins as a sequential program by executing one statement at a time. However, when the computer reaches a concurrent statement like this

```
cobegin 1 do
  input(name('prefix'))
also 2 do
  output
end
```

it calls the input and output procedures simultaneously. When both procedure calls are finished, the program continues as a sequential program after the concurrent statement (until it reaches another concurrent statement). In this way, an Edison program can alternate between sequential and concurrent modes of execution.

Using when–statements, you can implement a variety of well-known synchronization mechanisms including locks, semaphores, message buffers, and monitors.

Process Synchronization

A message buffer is represented by two variables

```
var slot: char; full: bool
```

The variable named slot holds the current contents of the buffer (a single character). The other variable has a boolean value that defines whether or not there is any character in the slot.

When the input process wants to send a character it must wait until the buffer is empty (otherwise, it will overwrite the previous character before it has been received). The process does this by means of a when statement

```
when not full do
  slot := value; full := true
end
```

If the buffer already is empty, the input process can immediately put a character into it. However, if the buffer is full, the input process must wait until the output process has received the previous character and made the buffer empty. The

when-statement automatically handles this scheduling problem by delaying the input process (if necessary) until the boolean expression *not* full is true. (If the input process is delayed, the output process can run in the meantime.) When the above condition is satisfied, the input process puts the character in the buffer and indicates that the buffer is full.

The output process waits until the buffer is full before receiving a character

```
when full do
    value:= slot; full := false
end
```

Afterwards, it indicates that the buffer is now empty.

If the two processes try to evaluate the expressions (or execute the bodies) of the when-statements simultaneously, the computer will schedule the processes one at a time. So when-statements are executed one at a time as indivisible actions.

When-statements control all process synchronization and communication. By using when-statements you can implement a variety of well-known synchronization mechanisms including locks, semaphores, message buffers, and monitors.

Modularity

In our programming example, two processes use the same buffer. Now, if you make the slightest programming mistake in a concurrent program like this, the program starts acting in such an unpredictable way that testing becomes almost impossible ("racing conditions").

Instead of letting the processes operate directly on the buffer by means of when-statements, it is much safer to hide the buffer inside a program module and let the processes access it indirectly by calling two procedures named send and receive.

So, instead of letting the processes operate directly on the buffer by means of when-statements, it is much safer to hide the buffer inside a program module and let the processes access it indirectly only by calling two procedures names send and receive:

```
module

  var slot: char; full: bool

  *proc send(value: char)
  begin
    when not full do
      slot:= value; full:= true
    end
  end

  *proc receive(var value: char)
  begin
    when full do
      value := slot; full :=
          false
    end
  end

begin full := false end
```

The buffer variables declared in the module can be accessed inside the module but not outside it. The asterisks in front of the procedures show that they can be called outside the module. The net effect of this protection scheme is to make sure that processes only perform the well-defined send and receive operations on the buffer.

Even these correct operations will not work, unless the buffer is initialized correctly. The begin-end statement at the end of the module makes sure that this happens: when the program execution starts, this statement is executed before the processes can call the procedures.

It is remarkable that this protection of the buffer variables is enforced completely during the compilation of the program without the use of hardware mechanisms. However, the most important thing about a module is that it combines all aspects of a programming concept (such as a buffer) in one place. By dividing large programs into modules, you make it possible for others to study them in small, self-contained pieces.

The Edison System for the IBM Personal Computer shows how successful this method can be. The operating system itself is an Edison program of 1200 lines divided into 15 modules. The average module consists of six procedures of 14 lines each.

Library Programs and External Procedures

If you want to use a programming language to write an operating system, the language must make it possible for one program (the operating system) to call other programs (the user programs). The user programs must be able to call external procedures that are defined inside the operating system; otherwise, the operating system cannot provide services for the user programs.

Pascal does not have these features and is therefore not suitable for operating system design (unless you extend the language).

An Edison program is a precompiled procedure that can be called by other Edison programs. Let us look at an operating system that calls a program named edit. In the operating system, the edit program is declared as a library procedure of the form.

```
lib proc edit
[ load(name('edit')) ]
```

When the operating system calls this procedure, a function named load inputs a code file named edit from a diskette. The edit program is then executed as a procedure.

This simple idea must be refined a bit to be useful. The operating system must be able to pass some of its own procedures as parameters to the edit program to support its use of the terminal and the disk files. Because it is impractical to declare every user program as a separate library procedure in the operating system, it must also be possible to pass the name of a user program as a parameter to a single library procedure.

With these extensions, the declaration of a program inside the op-

erating system looks like this:

```
lib proc userprogram(
    programname: name;
    . . .
    proc openread (var file:
      stream; title: name);
    proc more(var file: stream):
      bool;
    proc read(var file: stream;
      var value: char);
    . . .
[ load(programname) ]
```

A Program

The program parameters must be declared at the beginning of a user program. So the program example we discussed earlier, looks like this:

```
proc example(
    programname: name;
    . . .
    proc openread(var file:
      stream; title: name);
    proc more(var file: stream):
      bool;
    proc read(var file: stream;
      var value: char);
    . . .
const etx = char(3)
module
    . . .
    *proc send(value: char)
    begin . . . end
    *proc receive(var value: char)
    begin . . . end
begin . . . end
proc input(title: name)
begin . . . end
proc output
begin . . . end
begin
    cobegin 1 do
      input(name('prefix'))
    also 2 do
      output
    end
end
```

Every Edison program is a precompiled procedure that can be called by other Edison programs.

The Compiler

Every Edison program is a precompiled procedure that can be called by other Edison programs. There is no difference between the operating system and the user programs. The operating system is merely the first program that is loaded when the computer is started.

The Edison compiler is written in Edison. It consists of a main program that calls four subprograms (called passes) one at a time. The first pass inputs a source text from a disk file, performs lexical analysis and outputs intermediate code in another disk file. The second pass reads the intermediate code, performs syntax and scope analysis and outputs intermediate code. The third pass performs type checking and the last pass generates code.

All the input/output is hidden in procedures inside the main program. When the main program calls a subprogram, it passes these procedures as parameters to the subprogram. The main program is in effect a simple operating system for its own subprograms.

The clear isolation of system-dependent details in one part of the compiler made it possible to move the compiler from the LSI-11 computer to the IBM-PC in a few days.

The Edison-PC System can compile the largest pass of the Edison compiler on an IBM-PC with 64K bytes of memory and a dual disk drive. At the rate of two lines/sec, the compilation of pass three (1500 lines) takes 13 min.

Kernel Procedures

When you turn the IBM-PC on the Edison kernel starts the operating system. And, just as the operating system can pass some of its own procedures as parameters to other programs, the kernel can pass some of its own procedures as parameters to the operating system.

The kernel procedures perform basic input/output operations, such as "accept a character from the keyboard" or "output a sector on a disk." Kernel procedures provide a relatively safe method of using assembly language procedures in Edison programs as the compiler is able to check the number and types of the parameters in each procedure call.

The operating system may pass some of the kernel procedures as parameters to user programs. The latter cannot tell the difference between operating system procedures and kernel procedures. In general, an Edison program can pass a mixture of its own procedures and its external procedures as parameters to other Edison programs.

The flexibility of Edison makes it possible to write non-trivial software systems as a collection of small programs that call one another. ■

Per Brinch Hansen is Henry Salvatori Professor of Computer Science at the University of Southern California and an expert in programming language technology and concurrent systems. He developed Concurrent Pascal in 1975 and invented the monitor concept. He has written Operating System Principles *(1973) and* The Architecture of Concurrent Programs *(1977), both published by Prentice-Hall.*

Size and Performance

The Edison-PC System consists of about 10,000 lines of program text:

Operating system	**1200 lines**
Edison compiler	**4200 lines**
Assembler	**1500 lines**
Editor	**600 lines**
Text formatter	**500 lines**
Other programs	**300 lines**
Edison programs	**8300 lines**
Edison kernel	**2100 lines**
Edison system	**10400 lines**

The following execution times of some of the operating system procedures were measured on the IBM Personal Computer. They include disk access times:

Create + Delete File	**100 msec**
Open + Close File	**20 msec**
More + Read Character	**2.4 msec**

Accessing the SERIAL PORT From COBOL

How COBOL can— with the aid of a few relatively simple assembly language subroutines— perform an ASCII file transfer from the IBM PC to another computer system.

TOM CARTER

COBOL seems to be a forgotten language on the IBM PC. Practically every software package I read or hear about has been implemented using one form of BASIC or another. With the volumes of software systems already existing in COBOL this doesn't seem to make sense.

Why is BASIC the language of choice on a microcomputer? I think the answers are fairly obvious.

The growth of the microcomputer industry started with systems much less capable than the ones that are available today. The limited hardware and software capabilities of the early systems necessitated the utilization of an interpretive language like BASIC. No compiling or linking was required. Users found (and find) BASIC a simple to learn and use language. In addition, most of the earlier software packages were limited in scope. There weren't any full-blown financial packages as there are today.

Somewhere along the way the precedent was set. BASIC became the predominately used microcomputer language.

Today, we find microcomputers competing with minicomputers. Most of the past microcomputer limitations have been eliminated. We can buy a PC with more than a half a megabyte of memory and with enough, relatively fast mass storage to satisfy almost every use.

Still, COBOL is only slowly being adopted as a microcumputer language. This is partially due to the additional memory and mass storage costs required for any compiled language. This additional cost would also be incurred when using a compiled version of BASIC.

Large companies have also been very cautious in considering a microcomputer as a solution to any of their problems. As these companies begin using microcomputers, COBOL will become a dominant language as it is in most other business-related software systems. The

years of COBOL expertise will not be sacrificed. The existing software won't either. A way to salvage the technology and even the software will be found.

As large companies begin using microcomputers, cobol will become a dominant language as it is in most other business-related software systems. The years of cobol expertise will not be sacrificed.

COBOL Versus BASIC

In terms of execution speed, every compiled language is superior to every interpretive language. However, many merits of a language go beyond performance capabilities. In most cases, ease-of-use and structure are more important than microseconds. If this weren't the case, each of us would still be coding machine language.

It's unfair to compare interpretive BASIC to compiled COBOL in terms of performance. We would have to compare a compiled version of BASIC to a compiled version of COBOL to make this performance judgment. In fact, I don't know of such a comparison; the results would be very interesting.

I don't think an ease-of-use judgment can be made either. Both languages are equally easy to use in the hands of experts. Ease-of-use is a personal, subjective matter hard to assess.

The ability to structure a program, from my own experience, is the edge COBOL has on BASIC. You were better off not commenting or structuring BASIC programs because of the prices paid in terms of memory and speed. The first thing done when a BASIC program approaches the limitations of memory has typically been to discard all comments and descriptive identifiers.

In fact, BASIC lends itself to structuring just as well as COBOL. It just hasn't been used that way. I'm sure programs being developed in BASIC today look very much like the programs developed before. By this I mean instruction lines with multiple instructions, no comments, and very vague descriptive identifiers. This is fine if you're programming and maintaining for yourself. When programming for someone else and the possibility exists that another person will maintain and modify the program, then structure and simplicity must be a part of the implementation.

Importing Existing COBOL Software

There are many schemes for importing foreign software into a new computer system. Among the easier and less costly schemes involves transferring source files from one system to another. With most high level languages so standardized today, the conversion task becomes at worst a manual editing of each imported source file and at best an automated and programmed massaging of each file. Many language conversion tools are available for such tasks.

As a sample file transfer facility, I have prepared the following description and the following COBOL control program. Though the program does not conform in the strictest sense to normal COBOL structuring, it does provide a relatively good picture of how readable COBOL can be.

In the example, the COBOL program controls the flow of the system even though it never actually opens, reads, or closes a file. It doesn't even actually transfer the file to the other system. These functions are provided by assembly language routines that are linked with the COBOL program. These facilities might normally be found in implementations of COBOL, but they were not available in the version used.

Basic lends itself to structuring just as easily as cobol. It just hasn't been used that way.

The COBOL program (see listing 1) is only an example. The assembly language routines named OPFILE and RDFILE, which are referred to in the description, do not really exist. ASPORT (see listing 2) is an actual assembly language program developed as a general purpose runtime routine for COBOL.

In the example, ASPORT is utilized to transfer source data files from an IBM PC to another, unknown computer system. ASPORT could just as easily have been used to send an inquiry to a mainframe or to respond to an inquiry from a mainframe.

COBOL File Transfer Control Software

The following paragraphs describe and illustrate how COBOL, coupled with a few relatively simple assembly language subroutines, can perform an ASCII file transfer from the IBM PC to another computer system. The asynchronous port is the connecting link between the two systems.

The following assumptions and rules are mutually made and agreed upon by the control software modules of the two systems.

1. The file to be transferred from the IBM PC to the receiving computer system will be an ASCII data file containing no control characters other than carriage returns and line feeds. All other characters will be displayable ASCII characters.

Tom Carter's company, Real Time Concepts, develops utility, networking, and telecommunications software for the IBM PC and other computers.

Table 1: COBOL Working-storage Areas

Sample COBOL Status Packet:

000100	**01**	*Status-Packet.*		
000110		*02*	*Request-Indicator*	*Pic X.*
000120		*02*	*Return-Status*	*Pic X.*
000130		*02*	*Line-Status*	*Pic X.*

Sample COBOL Initialization Packet:

000140	**01**	*Initialization Packet.*		
000150		*02*	*Baud-Rate*	*Pic X (4).*
000160		*02*	*Parity*	*Pic X.*
000170		*02*	*Stop-Bits*	*Pic X.*
000180		*02*	*Data-Bits*	*Pic X.*

Sample Write Data Packet:

000190	**01**	*Output-Data-Buffer.*		
000200		*02*	*Output-Data-Count*	*Comp-0.*
000210		*02*	*Output-Data*	*Pic X (?).*

Sample Read Data Packet:

000220	**01**	*Input-Data-Buffer.*		
000230		*02*	*Input-Data-Count*	*Comp-0.*
000240		*02*	*Input-Terminator-1*	*Pic X.*
000250		*02*	*Input-Terminator-2*	*Pic X.*
000260		*02*	*Input-Data*	*Pic X (?).*

2. The file to be transferred will be a standard PC DOS sequential file. Each record in such a file is terminated by a carriage return and a line feed (ODH and OAH). The End-of-File (EOF) is signified by the existence of a 1AH character following the last record.

3. The protocol or handshaking procedures define exactly how data and control are to be transferred. The following protocol provisions are defined and utilized in our example COBOL control program and the software residing on the other computer system:

a. No file record will be greater than 120 characters in length. This includes the terminating carriage return and line feed characters. Therefore, the maximum data transfer will not exceed this length.

b. Each data transfer will be terminated by a line feed character. This allows the COBOL control software to read and transfer a data file record without modification or manipulation.

c. The receiving control software will postively or negatively acknowledge each data transfer. This means that the IBM PC COBOL control software must read the acknowledgement immediately following a record transfer.

d. These acknowledgements will be four characters in length. The first two characters will be a numeric status value with "00" signifying successful completion. Other status values will define various error conditions. The last two characters will be the standard carriage return and line feed terminators.

e. The reception of an acknowledgement will dictate to the COBOL control program what its next action should be. A "00" status will say, in effect, that the record has been received and that the receiving system is again prepared to read and process a next record. Other status values will dictate various error recovery and/or termination procedures.

COBOL File Transfer Control Software Overview

File transfer control programs are relatively simple in scope. When problems occur they are usually caused by modem or telephone connections or by confused handshaking procedures between the two connected computer systems.

The following steps define the essential ingredients found in a file transfer mechanism. The computer transferring the file is known as the SENDER and the computer receiving it is the RECEIVER.

Sender Program Flow

1. The SENDER control program must in some way ascertain the name of the file to be transferred.

2. This file must be opened and prepared for reading.

3. The communications interface must also be prepared for access. Included in this preparation are any actions required to make the connection to the other computer system. On the IBM PC these actions involve defining the various transmission characteristics of the asynchronous interface. Baud rate, parity, stop bits and data bits must be defined. If modems are used, some actions to dial and connect to the other system must be taken. Under program control, intelligent modems are capable of automating these actions.

When file transfer problems occur, they are usually caused by modem or telephone connections or by confused handshaking procedures between the two connected computer systems.

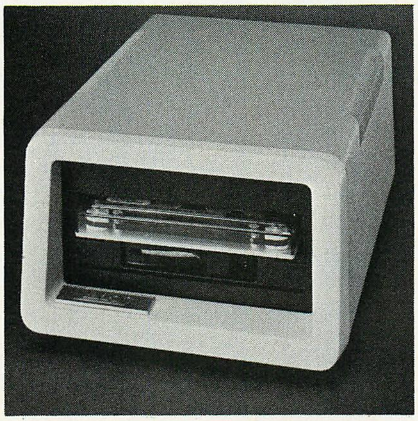
EVEN AN IBM PC® NEEDS A SAFETY NET.

Only Alloy's PC-Backup™ gives you high performance storage/retrieval, and protection – at a lower cost.

Now that the IBM PC comes with a 10 Mbyte hard disk, you need a Backup that can keep up with it. Alloy's PC-Backup is designed to do just that – and it beats flipping floppies every time.

Alloy's PC-Backup works like your business does: it is file-oriented, backing up files in an organized way. It is time and cost efficient too, because it never lets you lose important data.

Designed exclusively for the IBM PC, and small to large businesses, here are some of the features that make Alloy's PC Backup the finest safety net PC owners could ever require:



- Winchester hard disk backup and restore
- Comprehensive menu driven software package (Alloy's industry standard Tape Interchange Package)
- High speed data transfers
- Up to 16.5 Mbytes of data per cartridge

At your ISO now for only $1,995.

Also available, Alloy's new 9-track tape drive, for mainframe compatibility.

Alloy Computer Products
100 Pennsylvania Avenue
Framingham, Mass. 01701
(617) 875-6100

IBM PC is a registered trademark of IBM Corp. PC-Backup is a trademark of Alloy Computer Products.

CIRCLE NO. 226 ON READER SERVICE CARD

ALLOY
Computer Products

Table 2: ASPORT Function and Status Codes

Function Codes

I = initialize
W = write
R = read

Return-Status Codes

Y = successful function completion
N = unsuccessful function completion
I = invalid parameters

Line-Status Error Codes

When RETURN-STATUS contains "I":

1 = invalid function
2 = invalid baud-rate parameter
3 = invalid parity parameter
4 = invalid stop-bits parameter
5 = invalid data-bits parameter

When RETURN-STATUS contains "N":

1 = overrun error
2 = parity error
3 = framing error
4 = break detected
5 = data set not ready
6 = multiple errors
7 = time out
8 = data set not ready and no clear to send
9 = end-of-buffer

Table 3: Initialization-Packet Values

Baud-Rate

100	=	*110 baud*
150	=	*150 baud*
300	=	*300 baud*
600	=	*600 baud*
1200	=	*1200 baud*
2400	=	*2400 baud*
4800	=	*4800 baud*
9600	=	*9600 baud*

Note: *All 3 character BAUD-RATE specifications must be left-justified and space-filled.*

For example:
MOVE '300' to BAUD-RATE.

Parity

E = even parity
O = odd parity
N = no parity specification

Stop-Bits

1 = 1 stop bit
2 = 2 stop bits

Data Bits

7 = seven data bits
8 = eight data bits

4. Once the preparation tasks have been completed, the SENDER control program reads and transfers each record of the source file to the RECEIVER computer system. The complexity of this task generally depends on the protocol (handshaking) procedures that have been adopted by the computer systems. A protocol defines how errors are detected and how the control of the interface occurs.
5. After all file records have been read and transferred successfully, the SENDER system must notify the RECEIVER system so that the received file can be closed and preserved.
6. Once this notification has occurred, the SENDER system closes the source file and executes its next action. This next action might be program termination or might be preparation for the next file transfer.

Receiver Program Flow

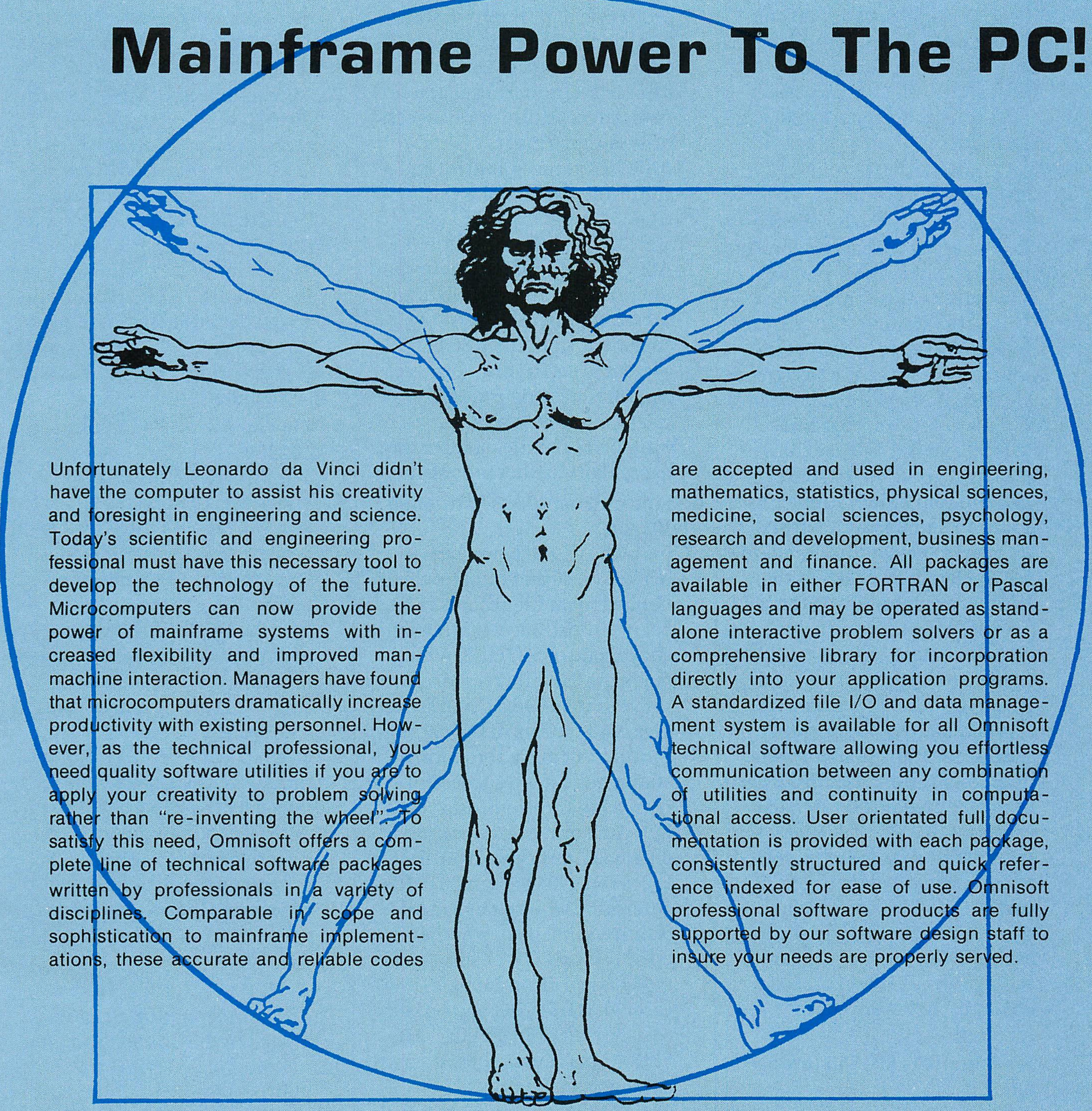
1. The RECEIVER control program must prepare its communications interface.
2. The file being transferred must be identified to the RECEIVER system. In the case of the following example COBOL SENDER program, the file name is transferred to the RECEIVER before any of the actual file data. This allows the RECEIVER program to create and prepare a file by the same name for the file data being transferred. In other cases, the RECEIVER program might just use a scratch file for this purpose and then rename the file after the successful completion of the transfer.
3. Once the RECEIVER file has been created and prepared, the RECEIVER control program simply reads each file record from the asynchronous port and then writes the record to the newly created file. This continues until all records have been received and written to the file successfully.
4. The created file is then closed and preserved on the RECEIVER system.

Line by Line Description of Sample COBOL Program

The following paragraphs describe on a line by line basis how the sample COBOL program successfully implements the SENDER side mechanisms in the file transfer interface.

1. **Line numbers 001070 through 001090**
   The system operator is prompted for the name of the file to be transferred and the name is accepted.
2. **Line numbers 001110 through 001160**
   The file is opened and the completion status is checked. A file open error status is handled appropriately. Notice that the COBOL program does not open the file itself since dynamic file allocation is difficult or non-existent in most COBOL implementations. Assembly language routine OPFILE is used for this purpose. The COBOL program passes a status area and an area containing the file name to OPFILE which opens the file and returns a completion status to the calling COBOL program.
3. **Line numbers 001220 through 001310**
   Once the file has been successfully opened, the asynchronous port is the next resource to be handled. It must be initialized. This includes defining its operating characteristics. The baud rate is set to 9600 baud since the assumed connection to the other system is a direct connection. No modems are involved. There is no parity and there is 1 stop bit and 7 data bits in each character. An "I" (initialize) is placed in the request indicator and the call to the asynchronous Assembly language interface routine is

Omnisoft Brings
Mainframe Power To The PC!



Unfortunately Leonardo da Vinci didn't have the computer to assist his creativity and foresight in engineering and science. Today's scientific and engineering professional must have this necessary tool to develop the technology of the future. Microcomputers can now provide the power of mainframe systems with increased flexibility and improved man-machine interaction. Managers have found that microcomputers dramatically increase productivity with existing personnel. However, as the technical professional, you need quality software utilities if you are to apply your creativity to problem solving rather than "re-inventing the wheel". To satisfy this need, Omnisoft offers a complete line of technical software packages written by professionals in a variety of disciplines. Comparable in scope and sophistication to mainframe implementations, these accurate and reliable codes are accepted and used in engineering, mathematics, statistics, physical sciences, medicine, social sciences, psychology, research and development, business management and finance. All packages are available in either FORTRAN or Pascal languages and may be operated as stand-alone interactive problem solvers or as a comprehensive library for incorporation directly into your application programs. A standardized file I/O and data management system is available for all Omnisoft technical software allowing you effortless communication between any combination of utilities and continuity in computational access. User orientated full documentation is provided with each package, consistently structured and quick reference indexed for ease of use. Omnisoft professional software products are fully supported by our software design staff to insure your needs are properly served.

Numerical methods • Statistical analysis • Engineering and Business Graphics
Control Systems Analysis and Design • Dynamic Systems Modeling and Analysis
Structural Analysis and Design • Thermodynamics • Heat Transfer
Fluid Mechanics • Optics • Mechanical, Civil, Chemical, Electrical Engineering

Omnisoft is constantly developing new software tools for the technical professional. Write today for detailed descriptions of Omnisoft technical software products and services.

OMNISOFT CORPORATION
9960 Owensmouth Avenue, Suite 32
Chatsworth, CA 91311 • (213) 709-1286
Dept. T1

CIRCLE NO. 162 ON READER SERVICE CARD

executed. Notice that no status checking occurs at the return point in the COBOL program. This is simply because no consistent method was found for failure determination. If the initiation fails, the first access to the asynchronous port will echo the failure.

4. **Line numbers 001370 through 001520**
Once the file has been opened and the interface has been initialized, the SENDER program transfers the file name to the RECEIVER system. A "W" request indicator directs the asynchronous runtime routine to transfer the data. Notice in the building of the output data that lines 001430 and 001440 use the "13" and "10" notation. In some COBOL implementations this notation defines the 13th and 10th characters of the standard ASCII character chart. In the sample program, this is the way that the carriage return and line feed characters are moved, as the terminator characters, into the output buffer. Notice also the mechanism used to initialize the output transfer count. The implementation of COBOL used in the testing of ASPORT had the bytes of a COMP-0 data item reversed from the way expected by the assembly language routine. ASPORT expected the most significant bits in the left byte and the least significant in the right byte. When passed by COBOL, they were exactly reversed.

5. **Line numbers 001580 through 001750**
After successful transmission of the file name, the SENDER system must receive and interrogate the acknowledgement from the RECEIVER system. To do this, the SENDER constructs and passes a status area and an input reception area (4 characters in length with a line feed character specified as the data terminator) to ASPORT. ASPORT uses the input reception buffer to receive and store the acknowledgement. Remember, the first two characters contain a two digit numeric status code and the last two characters contain the carrige return and line feed terminating characters. After successfully or unsuccessfully completing its task, ASPORT returns control and a status indicator to the calling COBOL program. Lines 001710 and 001720 interrogate the ASPORT completion status and lines 001740 and 001750 interrogate whether the acknowledgement from the RECEIVER system is positive or negative. Appropriate error routines might attempt to retry the transmission of the file name or, perhaps, abort further processing.

6. **Line numbers 001870 through 001940**
Now the COBOL program is ready to read and transfer the individual data file records. This begins by calling assembly language routine RDFILE, which reads the data file records, stores them in the COBOL output data area, and returns to the calling routine. A binary zero return status indicates a successful read, a binary 1 indicates the EOF, and other status values indicate various types of errors. Lines 001890 through 001910 determine whether a read error occurred and, if so, whether the error was an EOR or some other type of read error. Lines 001930 through 001940 separate a successful read from an EOF. An EOF status causes several, critical functions to occur:
   a. The EOF condition must be made known to the RECEIVER system for proper file closing.
   b. The file on the SENDER system should be closed.
   c. The next action, whether it be another data file transfer or program termination, occurs. Runtime routine RDFILE does more than simply read the file record and place into the COBOL transmission buffer. It also guarantees the existence of the record terminators and it places the character count of the record into the transmission byte counter for the asynchronous interface routine.

7. **Line numbers 002000 through 002070**
Once a record is successfully read, it is transferred (written) to the other computer system. The COBOL program accomplishes this by giving the file read buffer (where routine RDFILE placed the record) to ASPORT for the actual data transfer. An error return status causes the execution of appropriate error handling procedures.

8. **Line numbers 002130 through 002300**
The positive or negative acknowledgement is read from the RECEIVER computer system. A positive acknowledgement states that the record was received and written to the file and that the RECEIVER is again prepared to receive a data record. A negative acknowledgement states that an error has occurred. The specific status would dictate recovery, retry, and/or termination procedures to be followed by the SENDER system. After receiving a positive acknowledgement, the SENDER program returns to the top of the processing loop to read and transfer the next record.

ASPORT

The remainder of this article is dedicated to the formats and interfacing rules of ASPORT, the COBOL asynchronous interface runtime routine. These are included for clarity and informative purposes.

Return Status Values

The STATUS-PACKET (see table 1) is used by the COBOL program to indicate the type of request (initialize, write, or read) being made to runtime routine ASPORT and is used by ASPORT to return the completion status of the request to

calling COBOL program. The calling COBOL program must initialize STATUS-PACKET entry REQUEST-INDICATOR to the function request code before calling ASPORT. Table 2 shows the function and status codes, and the conditions which generate them.

Asynchronous Port Initialization

The asynchronous port must be initialized before any data reading or data writing can occur. To initialize the asynchronous port, the calling COBOL program must initialize the STATUS-PACKET and INITIALIZATION-PACKET areas before calling runtime routine ASPORT.

The STATUS-PACKET defines the requested function (INITIALIZE) to ASPORT and provides an area to be used by ASPORT to return two completion status values (RETURN-STATUS and LINE-STATUS) to the calling COBOL program. To initialize these parameters, move an "I" to variable REQUEST-INDICATOR and move low-values to varibles RETURN-STATUS and LINE-STATUS.

The INITIALIZATION-PACKET defines the various port variables to ASPORT and must also be initialized by the calling COBOL program. Table 3 defines the acceptable values for the variables in the INITIALIZATION-PACKET.

Upon completion of the initialization request, ASPORT moves a completion status to entry RETURN-STATUS and moves an error status (see TECHNICAL REFERENCE, appendix A pages 20 and 21) to entry LINE-STATUS. The COBOL program is responsible for interpreting, verifying, and handling these completion status values.

A sample initialization request is shown in listing 2.

Asynchronous Port Write

To write data to the asynchronous port, the calling COBOL program must initialize the STATUS-PACKET and OUTPUT-DATA-BUFFER areas before calling ASPORT. The STATUS-PACKET defines the requested function (WRITE) to ASPORT and provides an area to be used by ASPORT to return two completion status values (RETURN-STATUS and LINE-STATUS) to the calling COBOL program. To initialize these parameters, move a "W" to variable REQUEST-INDICATOR and move low-values to variables RETURN-STATUS and LINE-STATUS.

The OUTPUT-DATA-BUFFER defines the count of characters to be sent (written) in variable OUTPUT-DATA-COUNT and the actual data characters in user-defined variable OUTPUT-DATA. The calling COBOL program is responsible for initializing these variables. ASPORT uses the OUTPUT-DATA-COUNT to control the number of data characters written to the asynchronous port.

If an error does not occur during the transfer, ASPORT will return to the calling COBOL program after all data characters have been transferred. STATUS-PACKET variable RETURN-STATUS will be "Y" and STATUS-PACKET variable LINE-STATUS will be "O" (successful completion). If an error does occur during transmission, ASPORT will cease sending data characters, setup STATUS-PACKET variable RETURN-STATUS to "N," setup STATUS-PACKET variable LINE-STATUS to the appropriate error code (refer to RETURN-STATUS VALUES) and then return to the calling COBOL program.

The hardware architecture of the IBM PC represents 16 bit binary numbers with the most significant eight bits of the number in the left byte of a word and the least significant eight bits in the right byte. Initial testing of ASPORT found that COBOL represents a 16 bit binary data item (COMP-0) in the reverse order. It became necessary for COBOL to reverse COMP-0 data items before calling ASPORT. The example in listing 3 illustrates this point.

Asynchronous Port Read

To read data from the asynchronous port, the calling COBOL program must initialize the STATUS-PACKET and INPUT-DATA-BUFFER areas before calling ASPORT. The STATUS-PACKET defines the requested function (READ) to ASPORT and provides an area to be used by ASPORT to return two completion status values (RETURN-STATUS and LINE-STATUS) to the calling COBOL program. To initialize these parameters, move an "R" to variable REQUEST-INDICATOR and move low-values to variables RETURN-STATUS and LINE-STATUS.

The INPUT-DATA-BUFFER is subdivided into four entries:

INPUT-DATA-COUNT
INPUT-TERMINATOR-1
INPUT-TERMINATOR-2
INPUT-DATA

INPUT-DATA-COUNT must be initialized by the calling COBOL program to equal the length (in characters) of the INPUT-DATA entry. ASPORT uses this numeric entry to insure that the input data area is not overrun (more data than space).

INPUT-TERMINATOR-1 and INPUT-TERMINATOR-2 must be initialized by the calling COBOL program to the 1 or 2 character end-of-transmission indentifier. If no end-of-transmission identifier exists, then a space must be moved to each terminator entry. If only one end-of-transmission identifier exists, then that identifier must be moved to INPUT-TERMINATOR-1 and a space must be moved to INPUT-TERMINATOR-2.

The INPUT-DATA entry is used by ASPORT to save the received input characters. The INPUT-DATA-COUNT entry defines the size (how many characters will fit) of the INPUT-DATA entry. As characters are received, ASPORT checks each against the specified end-of-transmission characters. If the end-of-transmission characters are received, ASPORT saves them in the INPUT-DATA area and places the count of characters (in-

cluding the end-of-transmission characters) in the INPUT-DATA-COUNT entry.

If no end-of-transmission characters are specified, ASPORT reads data until the INPUT-DATA entry is full (based on the INPUT-DATA-COUNT). Again, the INPUT-DATA-COUNT entry is set to reflect the actual count of characters received and held in the INPUT-DATA area.

Refer to RETURN STATUS VALUES for a description of the return status conditions. If an end-of-transmission identifier is specified but not received before the end-of-buffer condition, then RETURN-STATUS will be set to "N" and LINE-STATUS will be set to "9."

An example of a read request is shown in listing 4.

The program ASPORT is shown in listing 5. ■

TECH JOURNAL PROGRAM LISTING #1

```
000100************************************************************
000110*     D A T A   D I V I S I O N
000120************************************************************
000130*
000140 DATA DIVISION.
000150*
000160 WORKING-STORAGE SECTION.
000170*
000180 01  FILE-NAME                     PIC X(16).
000190*
000200 01  OPEN-RETURN-STATUS.
000210     03  OPEN-STATUS-VALUE         COMP-O.
000220*
000230 01  READ-RETURN-STATUS.
000240     03  READ-STATUS-VALUE         COMP-O.
000250*
000260 01  STATUS-PACKET.
000270     03  REQUEST-INDICATOR         PIC X.
000280     03  RETURN-STATUS             PIC X.
000290     03  LINE-STATUS               PIC X.
000300*
000310 01  INITIALIZATION-PACKET.
000320     03  BAUD-RATE                 PIC X(4).
000330     03  PARITY                    PIC X.
000340     03  STOP-BITS                 PIC X.
000350     03  DATA-BITS                 PIC X.
000360*
000370 01  OUTPUT-DATA-BUFFER.
000380     03  OUTPUT-DATA-COUNT         COMP-O.
000390     03  OUTPUT-DATA               PIC X(120).
000400*
000410 01  INPUT-DATA-BUFFER.
000420     03  INPUT-DATA-COUNT          COMP-O
000430     03  INPUT-TERMINATOR-1        PIC X.
000440     03  INPUT-TERMINATOR-2        PIC X.
000450     03  INPUT-DATA-STATUS         PIC X(2).
000460     03  INPUT-DATA                PIC X(2).
000470*
000480 01  SWITCH-1.
000490 01  SWITCH-2                      REDEFINES SWITCH-1.
000500     03  SWITCH-HIGH-1             PIC X.
000510     03  SWITCH-LOW-1              PIC X.
000520*
000530 01  SWITCH-3.
000540 01  SWITCH-4                      REDEFINES SWITCH-3.
000550     03  SWITCH-HIGH-2             PIC X.
000560     03  SWITCH-LOW-2              PIC X.
000570*
000580 01  WRITE-FILE-PACKET.
000590     03  WRITE-FILE-COUNT          COMP-O.
000600     03  WRITE-FILE-NAME           PIC X(16).
000610     03  WRITE-FILE-CR             PIC X.
000620     03  WRITE-FILE-LF             PIC X.
000630*

001000************************************************************
001010*     P R O C E D U R E   D I V I S I O N
001020************************************************************
001030 PROCEDURE DIVISION.
001040*
001050 0100-BEGIN-FILE-TRANSFER.
001060*
001070     MOVE SPACES TO FILE-NAME.
001080     DISPLAY "ENTER NAME OF FILE = ".
001090     ACCEPT FILE-NAME.
001100*
001110     MOVE ZERO TO OPEN-STATUS-VALUE.
001120*
001130     CALL "OPFILE" USING OPEN-RETURN-STATUS, FILE-NAME.
001140*
001150     IF OPEN-STATUS-VALUE NOT EQUAL ZERO
001160         PERFORM 0900-FILE-OPEN-ERROR.
001170*
001180************************************************************
001190*    INITIALIZE THE ASYNCHRONOUS PORT
001200************************************************************
001210*
001220     MOVE "1200" TO BAUD-RATE.
001230     MOVE "N" TO PARITY.
001240     MOVE "1" TO STOP-BITS.
001250     MOVE "7" TO DATA-BITS.
001260*
001270     MOVE "I" TO REQUEST-INDICATOR.
001280     MOVE LOW-VALUES TO RETURN-STATUS.
001290     MOVE LOW-VALUES TO LINE-STATUS.
001300*
001310     CALL "ASPORT" USING STATUS-PACKET, INITIALIZATION-PACKET.
001320*
001330************************************************************
001340* WRITE THE FILE NAME TO THE OTHER SYSTEM
001350************************************************************
001360*
001370     MOVE "W" TO REQUEST-INDICATOR.
001380     MOVE LOW-VALUES TO RETURN-STATUS.
001390     MOVE LOW-VALUES TO LINE-STATUS.
001400*
001410     MOVE SPACES TO WRITE-FILE-NAME.
001420     MOVE FILE-NAME TO WRITE-FILE-NAME.
001430     MOVE ""13"" TO WRITE-FILE-CR.
001440     MOVE ""10"" TO WRITE-FILE-LF.
001450     MOVE 18 TO SWITCH-1.
001460     PERFORM 0950-SWITCH-BINARY-NUMBERS.
001470     MOVE SWITCH-3 TO WRITE-FILE-COUNT.
001480*
001490     CALL "ASPORT" USING STATUS-PACKET, WRITE-FILE-PACKET.
001500*
001510     IF RETURN-STATUS NOT EQUAL "0"
001520         PERFORM 0960-ASYNCHRONOUS-WRITE-ERROR.
001530*

001540************************************************************
001550*    READ ACKNOWLEDGEMENT FROM CONNECTED SYSTEM
001560************************************************************
001570*
001580     MOVE "R" TO REQUEST-INDICATOR.
001590     MOVE LOW-VALUES TO RETURN-STATUS.
001600     MOVE LOW-VALUES TO LINE-STATUS.
001610*
001620     MOVE SPACES TO INPUT-DATA, INPUT-DATA-STATUS.
001630     MOVE SPACES TO INPUT-TERMINATOR-2.
001640     MOVE ""10"" TO INPUT-TERMINATOR-1.
001650     MOVE 4 TO SWITCH-1.
001660     PERFORM 0950-SWITCH-BINARY-NUMBERS.
001670     MOVE SWITCH-3 TO INPUT-DATA-COUNT.
001680*
001690     CALL "ASPORT" USING STATUS-PACKET, INPUT-DATA-BUFFER.
001700*
001710     IF RETURN-STATUS NOT EQUAL "0"
001720         PERFORM 0970-ASYNCHRONOUS-READ-ERROR.
001730*
001740     IF INPUT-DATA-STATUS NOT EQUAL "00"
001750         PERFORM 0970-ASYNCHRONOUS-READ-ERROR.
001760*
```

```
*************************************************************
*    READ AND TRANSFER FILE RECORD TO CONNECTED SYSTEM
*************************************************************
*
 0200-READ-AND-TRANSFER-LOOP.
*
*************************************************************
*    READ THE FILE RECORD FROM THE FILE
*************************************************************
*
     CALL "RDFILE" USING READ-RETURN-STATUS, OUTPUT-DATA-BUFFER.
*
     IF READ-STATUS-VALUE NOT EQUAL ZERO
         IF READ-STATUS-VALUE NOT EQUAL 01
             PERFORM 0910-FILE-READ-ERROR.
*
     IF READ-STATUS-VALUE EQUAL 01
         PERFORM 0920-END-OF-DATA-FILE.
*
*************************************************************
*    WRITE THE FILE RECORD TO THE CONNECTED SYSTEM
*************************************************************
*
     MOVE "W" TO REQUEST-INDICATOR.
     MOVE LOW-VALUES TO RETURN-STATUS.
     MOVE LOW-VALUES TO LINE-STATUS.
*
     CALL "ASPORT" USING STATUS-PACKET, OUTPUT-DATA-BUFFER.
*
     IF RETURN-STATUS NOT EQUAL ZERO
         PERFORM 0960-ASYNCHRONOUS-WRITE-ERROR.
*
*************************************************************
*    READ ACKNOWLEDGEMENT FROM CONNECTED SYSTEM
*************************************************************
*
     MOVE "R" TO REQUEST-INDICATOR.
     MOVE LOW-VALUES TO RETURN-STATUS.
     MOVE LOW-VALUES TO LINE-STATUS.
*
     MOVE SPACES TO INPUT-DATA, INPUT-DATA-STATUS.
     MOVE SPACES TO INPUT-TERMINATOR-2.
     MOVE ""10"" TO INPUT-TERMINATOR-1.
     MOVE 4 TO SWITCH-1.
     PERFORM 0950-SWITCH-BINARY-NUMBERS.
     MOVE SWITCH-3 TO INPUT-DATA-COUNT.
*
     CALL "ASPORT" USING STATUS-PACKET, READ-DATA-BUFFER.
*
     IF RETURN-STATUS NOT EQUAL "0"
         PERFORM 0960-ASYNCHRONOUS-READ-ERROR.
*
     IF INPUT-DATA-STATUS NOT EQUAL "00"
         PERFORM 0960-ASYNCHRONOUS-READ-ERROR.
*
     GO TO 0200-READ-AND-TRANSFER-LOOP.
*
```

TECH JOURNAL PROGRAM LISTING #2

```
SAMPLE ASYNCHRONOUS PORT INITIALIZATION REQUEST

*
 INITIALIZE-ASYNCHRONOUS-PORT.
*
     MOVE 'I' TO REQUEST-INDICATOR.
     MOVE LOW-VALUES TO RETURN-STATUS.
     MOVE LOW-VALUES TO LINE-STATUS.
*
     MOVE '1200' TO BAUD-RATE.
     MOVE 'E' TO PARITY.
     MOVE '1' TO STOP-BITS.
     MOVE '8' TO DATA-BITS.
*
     CALL 'ASPORT' USING STATUS-PACKET, INITIALIZATION-PACKET.
*
     IF RETURN-STATUS NOT EQUAL 'Y'
         PERFORM USER-ERROR-ROUTINE.
*
```

TECH JOURNAL PROGRAM LISTING #3

```
SAMPLE ASYNCHRONOUS PORT WRITE REQUEST

*
 01  SWITCH-1                   COMP-0.
 01  SWITCH-2                   REDEFINES SWITCH-1.
     02  SWITCH-HIGH-1          PIC X.
     02  SWITCH-LOW-1           PIC X.
*
 01  SWITCH-3                   COMP-0.
 01  SWITCH-4                   REDEFINES SWITCH-3.
     02  SWITCH-HIGH-2          PIC X.
     02  SWITCH-LOW-2           PIC X.
*

*
 WRITE-ASYNCHRONOUS-PORT.
*
     MOVE 'W' TO REQUEST-INDICATOR.
     MOVE LOW-VALUES TO RETURN-STATUS.
     MOVE LOW-VALUES TO LINE-STATUS.
*
     MOVE 14 TO OUTPUT-DATA-COUNT.
     PERFORM 0900-SWITCH-BINARY-NUMBERS.
     MOVE 'THIS IS A TEST' TO OUTPUT-DATA.
*
     CALL 'ASPORT' USING STATUS-PACKET, OUTPUT-DATA-BUFFER.
*
     IF RETURN-STATUS NOT EQUAL 'Y'
         PERFORM USER-ERROR-ROUTINE.
*

*
 0900-SWITCH-BINARY-NUMBERS.
*
     MOVE OUTPUT-DATA-COUNT TO SWITCH-1.
     MOVE SWITCH-HIGH-1 TO SWITCH-LOW-2.
     MOVE SWITCH-LOW-1 TO SWITCH-HIGH-2.
     MOVE SWITCH-3 TO OUTPUT-DATA COUNT.
     EXIT.
*
```

TECH JOURNAL PROGRAM LISTING #4

```
SAMPLE ASYNCHRONOUS PORT READ REQUEST

*
 READ-ASYNCHRONOUS-PORT.
*
     MOVE 'R' TO REQUEST-INDICATOR.
     MOVE LOW-VALUES TO RETURN-STATUS.
     MOVE LOW-VALUES TO LINE-STATUS.
*
     MOVE 80 TO INPUT-DATA-COUNT.
     PERFORM 0900-SWITCH-BINARY-NUMBERS.
     MOVE SPACE TO INPUT-TERMINATOR-1.
     MOVE SPACE TO INPUT-TERMINATOR-2.
     MOVE SPACES TO INPUT-DATA.
*
     CALL 'ASPORT' USING STATUS-PACKET, INPUT-DATA-BUFFER.
*
     IF RETURN-STATUS EQUAL 'N'
         PERFORM USER-ERROR-ROUTINE.
*
^Z
```

TECH JOURNAL PROGRAM LISTING #5

```
        PAGE 60,132
        PAGE
TITLE   ASPORT
;
;       (C)COPYRIGHT 1983 - REAL TIME CONCEPTS, INC.
;
;       AUTHOR: THOMAS V. CARTER
;
;       IBM PC ASYNCHRONOUS I/O INTERFACE
;
;       ALLOCATE THE WORKING STORAGE SEGMENT
;
WSSSEG  SEGMENT PARA PUBLIC 'DATA'
;
;       ALLOCATE THE VALID BAUD RATE SPECIFICATIONS
;
BDRATE  DB      '110 '
        DB      '150 '
        DB      '300 '
        DB      '600 '
        DB      '1200'
        DB      '2400'
        DB      '4800'
        DB      '9600'
;
WSSSEG  ENDS
;
        PAGE
;
;       ASYNCHRONOUS PORT INTERFACE ROUTINE
;
SUBSEG  SEGMENT PARA PUBLIC 'CODE'
;
ASPORT  PROC    FAR
        ASSUME  CS:SUBSEG               ;CS = CODE SEGMENT
        PUBLIC  ASPORT                  ;EXTERNAL SYMBOL
;
;       VALIDATE THE REQUESTED FUNCTION
;
        MOV     BX,ES                   ;BX = COBOL EXTRA SEGMENT
        PUSH    BX                      ;SAVE THE COBOL EXTRA SEGMENT
        MOV     BX,WSSSEG               ;BX = NEW EXTRA SEGMENT ADDRESS
        MOV     ES,BX                   ;ES = NEW EXTRA SEGMENT ADDRESS
        PUSH    BP                      ;SAVE BP FOR COBOL RETURN
        MOV     BP,SP                   ;BP = CURRENT STACK FRAME
        MOV     BX,[BP+10]              ;BX = ARGUMENT 1 ADDRESS
        MOV     AL,[BX]                 ;AL = FUNCTION REQUEST
;
;       CHECK FOR AN INITIALIZATION REQUEST
;
        CMP     AL,'I'                  ;IS THIS AN INITIALIZE REQUEST?
        JNE     ASV010                  ;BRANCH IF NOT AN INIT REQUEST
;
        JMP     ASINIT                  ;BRANCH TO INITIALIZE PORT
;
;       CHECK FOR A WRITE REQUEST
;
ASV010:
        CMP     AL,'W'                  ;IS THIS A WRITE REQUEST?
        JNE     ASV020                  ;BRANCH IF NOT A WRITE REQUEST
;
        JMP     ASSEND                  ;BRANCH TO WRITE DATA
;
;       CHECK FOR A READ REQUEST
;
ASV020:
        CMP     AL,'R'                  ;IS THIS A READ REQUEST?
        JNE     ASV030                  ;BRANCH IF NOT A READ REQUEST
;
        JMP     ASREAD                  ;BRANCH TO READ DATA
;
;       INVALID REQUEST FUNCTION // ERROR
;
ASV030:
        MOV     AH,'1'                  ;AH = 1 = INVALID FUNCTION
;
        PAGE
;
;       PARAMETER SPECIFICATION ERROR
```

```
;
ASPERR:
        MOV     AL,'I'                  ;AL = I = INVALID PARAMETERS
        MOV     BX,[BP+10]              ;BX = ARGUMENT 1 ADDRESS
        MOV     [BX+1],AL               ;SETUP RETURN STATUS
        MOV     [BX+2],AH               ;SETUP RETURN STATUS
;
;       RETURN TO COBOL CALLING ROUTINE
;
ASEXIT:
        POP     BP                      ;RESTORE BP FOR COBOL RETURN
        POP     BX                      ;RESTORE COBOL EXTRA SEGMENT
        MOV     ES,BX                   ;RESTORE COBOL EXTRA SEGMENT
        RET     4                       ;RETURN TO COBOL CALLER
;
        PAGE
;
;       ASYNCHRONOUS PORT INITIALIZATION
;
ASINIT:
        SUB     AX,AX                   ;AX = ZERO = BINARY BAUD RATE
        MOV     SI,AX                   ;SI = BAUD RATE TABLE INDEX
        MOV     BX,[BP+8]               ;BX = ARGUMENT 2 ADDRESS
        MOV     CX,[BX]                 ;CX = BAUD RATE SPECIFICATION
        MOV     DX,[BX+2]               ;DX = BAUD RATE SPECIFICATION

        MOV     BX,OFFSET BDRATE        ;BX = BAUD RATE TABLE ADDRESS
;
;       VALIDATE THE SPECIFIED BAUD RATE
;
ASI010:
        CMP     CX,ES:[BX+SI]           ;COMPARE TO BAUD RATE TABLE
        JNE     ASI020                  ;BRANCH IF NOT EQUAL
;
        CMP     DX,ES:[BX+SI+2]         ;COMPARE TO BAUD RATE TABLE
        JE      ASI030                  ;BRANCH IF VALID BAUD RATE
;
;       UPDATE POINTERS AND COUNTERS FOR NEXT ENTRY
;
ASI020:
        ADD     SI,4                    ;INCREMENT SI TO NEXT ENTRY
        INC     AX                      ;INCREMENT BINARY BAUD RATE
        CMP     AX,7                    ;COMPARE AX TO MAXIMUM COUNT
        JNG     ASI010                  ;BRANCH TO PROCESS NEXT ENTRY
;
        MOV     AH,'2'                  ;AH = 2 = INVALID BAUD RATE
        JMP     ASPERR                  ;BRANCH TO RETURN TO CALLER
;
        PAGE
;
;       VALIDATE THE PARITY SPECIFICATION
;
ASI030:
        MOV     CL,5                    ;CL = 5 = ROTATE COUNT
        ROL     AL,CL                   ;POSITION BAUD RATE BITS IN AL
        PUSH    AX                      ;SAVE BAUD RATE ON THE STACK
;
        MOV     BX,[BP+8]               ;BX = ARGUMENT 2 ADDRESS
        MOV     AL,[BX+4]               ;AL = PARITY SPECIFICATION
;
;       CHECK FOR EVEN PARITY
;
        CMP     AL,'E'                  ;EVEN PARITY?
        JNE     ASI040                  ;NOT EVEN PARITY // CHECK ODD
;
        MOV     AX,18H                  ;AX = 18H = EVEN PARITY
        JMP     ASI070                  ;BRANCH TO SETUP EVEN PARITY
;
;       CHECK FOR ODD PARITY
;
ASI040:
        CMP     AL,'O'                  ;ODD PARITY?
        JNE     ASI050                  ;NOT ODD PARITY // CHECK NONE
;
        MOV     AX,8H                   ;AX = 8H = ODD PARITY
        JMP     ASI070                  ;BRANCH TO SETUP ODD PARITY
;
;       CHECK FOR NO PARITY
;
ASI050:
        CMP     AL,'N'                  ;NO PARITY?
        JNE     ASI060                  ;NOT NO-PARITY // ERROR
;
        MOV     AX,0                    ;AX = ZERO = NO PARITY
        JMP     ASI070                  ;BRANCH TO SETUP NO PARITY
```

```
;
;       INVALID PARITY SPECIFICATION
;
ASI060:
        POP     DX                      ;RESTORE STACK POINTER
        MOV     AH,'3'                  ;AH = 3 = INVALID PARITY
        JMP     ASPERR                  ;BRANCH TO RETURN TO CALLER
;
        PAGE
;
;       VALIDATE THE NUMBER OF STOP BITS
;
ASI070:
        POP     DX                      ;RESTORE BAUD RATE
        OR      AX,DX                   ;OR THE BAUD RATE AND PARITY
        PUSH    AX                      ;SAVE RESULT ON THE STACK
;
        MOV     BX,[BP+8]               ;BX = ARGUMENT 2 ADDRESS
        MOV     AL,[BX+5]               ;AL = NUMBER OF STOP BITS
;
;       CHECK FOR ONE STOP BIT
;
        CMP     AL,'1'                  ;ONE STOP BIT?
        JNE     ASI080                  ;NOT ONE STOP BIT // CHECK TWO
;
        MOV     AX,0                    ;AX = 0 = 1 STOP BIT
        JMP     ASI100                  ;BRANCH TO SETUP STOP BITS
;
;       CHECK FOR TWO STOP BITS
;
ASI080:
        CMP     AL,'2'                  ;TWO STOP BITS?
        JNE     ASI090                  ;NOT TWO STOP BITS // ERROR
;
        MOV     AX,4H                   ;AX = 4 = TWO STOP BITS
        JMP     ASI100                  ;BRANCH TO SETUP STOP BITS
;
;       INVALID STOP BIT SPECIFICATION
;
ASI090:
        POP     DX                      ;RESTORE STACK POINTER
        MOV     AH,'4'                  ;AH = 4 = INVALID STOP BITS
        JMP     ASPERR                  ;BRANCH TO RETURN TO CALLER
;
        PAGE
;
;       VALIDATE THE NUMBER OF DATA BITS
;
ASI100:
        POP     DX                      ;RESTORE BAUD RATE AND PARITY
        OR      AX,DX                   ;OR ON THE STOP BITS
        PUSH    AX                      ;SAVE VALUES ON THE STACK
;
        MOV     BX,[BP+8]               ;BX = ARGUMENT 2 ADDRESS
        MOV     AL,[BX+6]               ;AL = NUMBER OF DATA BITS
;
;       CHECK FOR SEVEN DATA BITS
;
        CMP     AL,'7'                  ;SEVEN DATA BITS?
        JNE     ASI110                  ;NOT SEVEN // CHECK EIGHT
;
        MOV     AX,2                    ;AX = 2 = SEVEN DATA BITS
        JMP     ASI130                  ;BRANCH TO SETUP DATA BITS
;
;       CHECK FOR EIGHT DATA BITS
;
ASI110:
        CMP     AL,'8'                  ;EIGHT DATA BITS?
        JNE     ASI120                  ;NOT EIGHT DATA BITS // ERROR
;
        MOV     AX,3                    ;AX = 3 = EIGHT DATA BITS
        JMP     ASI130                  ;BRANCH TO SETUP DATA BITS
;
;       INVALID NUMBER OF DATA BITS
;
ASI120:
        POP     DX                      ;RESTORE STACK POINTER
        MOV     AH,'5'                  ;AH = 5 = INVALID DATA BITS
        JMP     ASPERR                  ;BRANCH TO RETURN TO CALLER
;
        PAGE
;
;       CALL BIOS TO INITIALIZE ASYNCHRONOUS PORT NUMBER ONE
;
ASI130:
        POP     DX                      ;RESTORE ACCUMULATED VALUES
        OR      AX,DX                   ;OR ON THE DATA BIT COUNT
        MOV     AH,0                    ;AH = 0 = INITIALIZE PORT
        MOV     DX,0                    ;DX = ZERO = 1ST ASYNCH PORT
        INT     14H                     ;CALL BIOS FOR INITIALIZATION
;
;       VALIDATE MODEM STATUS BITS // AL = STATUS BITS
;
        AND     AL,30H                  ;ISOLATE CRITICAL STATUS BITS
        CMP     AL,30H                  ;DATA SET READY // NO CLEAR
        JNE     ASI140                  ;BRANCH IF ERROR STATUS
;
;       SUCCESSFUL ASYNCHRONOUS PORT INITIALIZATION
;
        MOV     AL,'Y'                  ;SUCCESSFUL COMPLETION
        MOV     AH,'0'                  ;AH = 0 = SUCCESSFUL RETURN
        JMP     ASI150                  ;BRANCH TO RETURN TO CALLER
;
;       UNSUCCESSFUL ASYNCHRONOUS PORT INITIALIZATION
;
ASI140:
        MOV     AL,'Y'                  ;AL = Y = ERROR STATUS
        MOV     AH,'8'                  ;DATA SET NOT READY // NO CLEAR
;
;       SETUP RETURN STATUS VALUES AND RETURN TO CALLER
;
ASI150:
        MOV     BX,[BP+10]              ;BX = ARGUMENT 1 ADDRESS
        MOV     [BX+1],AL               ;SETUP RETURN STATUS
        MOV     [BX+2],AH               ;SETUP RETURN STATUS
        JMP     ASEXIT                  ;RETURN TO CALLER
;
        PAGE
;
;       ASYNCHRONOUS PORT WRITE REQUEST
;
ASSEND:
        MOV     BX,[BP+8]               ;BX = ARGUMENT 2 ADDRESS
        MOV     CX,[BX]                 ;CX = COUNT OF DATA CHARACTERS
        SUB     AX,AX                   ;AX = ZERO
        MOV     SI,AX                   ;SI = ZERO = DATA INDEX
        ADD     BX,2                    ;UPDATE BX TO START OF DATA
;
;       SEND EACH DATA CHARACTER TO THE ASYNCHRONOUS PORT
;
ASS010:
        MOV     AL,[BX+SI]              ;AL = OUTPUT DATA CHARACTER
        MOV     AH,1                    ;AH = 1 = WRITE DATA
        MOV     DX,0                    ;DX = 1ST ASYNCH PORT
        INT     14H                     ;CALL BIOS FOR WRITE
        MOV     AL,AH                   ;AH (AL) = COMPLETION STATUS
        AND     AL,80H                  ;ISOLATE STATUS FLAG (BIT 7)
        CMP     AL,0                    ;SUCCESSFUL DATA WRITE?
        JE      ASS030                  ;BRANCH IF SUCCESSFUL WRITE
;
;       UNSUCCESSFUL DATA TRANSFER // ERROR
;
        MOV     AL,'N'                  ;AL = N = ERROR STATUS
        MOV     AH,'7'                  ;AH = 7 = TRANSFER TIMEOUT
;
;       SETUP WRITE COMPLETION STATUS VALUES
;
ASS020:
        MOV     BX,[BP+10]              ;BX = ARGUMENT 1 ADDRESS
        MOV     [BX+1],AL               ;SETUP ERROR RETURN STATUS
        MOV     [BX+2],AH               ;SETUP ERROR RETURN STATUS
        JMP     ASEXIT                  ;RETURN TO CALLER
;
;       SUCCESSFUL OUTPUT TRANSFER // SEND MORE DATA?
;
ASS030:
        INC     SI                      ;INCREMENT THE DATA INDEX
        LOOP    ASS010                  ;SEND ALL DATA CHARACTERS
;
        MOV     AL,'Y'                  ;AL = Y = SUCCESSFUL COMPLETION
        MOV     AH,'0'                  ;AH = 0 = SUCCESSFUL COMPLETION
        JMP     ASS020                  ;BRANCH TO RETURN TO CALLER
;
        PAGE
;
;       ASYNCHRONOUS PORT READ REQUEST
```

```
;
ASREAD:
        SUB     CX,CX                   ;CX = ZERO = TERMINATOR SWITCH
        MOV     SI,CX                   ;SI = ZERO = INPUT BUFFER INDEX
        MOV     BX,[BP+8]               ;BX = ARGUMENT 2 ADDRESS
;
;       READ ONE CHARACTER AT A TIME FROM ASYNCHRONOUS PORT
;
ASR010:
        MOV     AH,2                    ;AH = 2 = READ ASYNCH PORT
        MOV     DX,0                    ;DX = ZERO = 1ST ASYNCH PORT
        INT     14H                     ;CALL BIOS FOR READ
        MOV     DH,AH                   ;DH (AH) = COMPLETION STATUS
        AND     DH,9EH                  ;ISOLATE ERROR STATUS BITS
        CMP     DH,0                    ;VERIFY COMPLETION STATUS
        JE      ASR020                  ;BRANCH IF SUCCESSFUL READ
;
;       ASYNCHRONOUS PORT READ ERROR // AH = STATUS
;
        MOV     [BX],SI                 ;SETUP INPUT CHARACTER COUNT
;
        MOV     AH,'1'                  ;AH = 1 = OVERRUN ERROR
        CMP     DH,02H                  ;OVERRUN ERROR?
        JE      ASR022                  ;BRANCH IF OVERRUN ERROR
;
        MOV     AH,'2'                  ;AH = 2 = PARITY ERROR
        CMP     DH,04H                  ;CHECK PARITY ERROR?
        JE      ASR022                  ;BRANCH IF PARITY ERROR
;
        MOV     AH,'3'                  ;AH = 3 = FRAMING ERROR
        CMP     DH,08H                  ;CHECK FRAMING ERROR
        JE      ASR022                  ;BRANCH IF FRAMING ERROR
;
        MOV     AH,'4'                  ;AH = 4 = BREAK DETECTED
        CMP     DH,10H                  ;CHECK BREAK DETECTED
        JE      ASR022                  ;BRANCH IF BREAK DETECTED
;
        MOV     AH,'5'                  ;AH = 5 = DATA SET NOT READY
        CMP     DH,80H                  ;CHECK DATA SET NOT READY
        JE      ASR022                  ;BRANCH IF DATA SET NOT READY
;
        MOV     AH,'6'                  ;AH = 6 = MULTIPLE ERRORS
;
;       SETUP READ ERROR STATUS // RETURN TO CALLER
;
ASR022:
        MOV     AL,'N'                  ;AL = N = ERROR STATUS
        JMP     ASS020                  ;BRANCH TO RETURN TO CALLER
;
        PAGE
;
;       STORE INPUT CHARACTER AND CHECK FOR END OF DATA
;
ASR020:
        MOV     [BX+SI+4],AL            ;STORE DATA IN INPUT BUFFER
        INC     SI                      ;INCREMENT INPUT BUFFER INDEX
        CMP     CX,0                    ;1ST TERMINATOR RECEIVED?
        JNE     ASR050                  ;YES - CHECK 2ND TERMINATOR
;
;       CHECK FOR 1ST DATA TERMINATOR
;
ASR030:
        MOV     DL,[BX+2]               ;DL = 1ST DATA TERMINATOR
        CMP     DL,20H                  ;IS THERE A 1ST TERMINATOR?
        JE      ASR060                  ;BRANCH IF NO TERMINATOR
;
        CMP     DL,AL                   ;IS THIS 1ST DATA TERMINATOR?
        JNE     ASR060                  ;BRANCH IF NOT 1ST TERMINATOR
;
;       1ST DATA TERMINATOR FOUND
;
        INC     CX                      ;INCREMENT CX - 2ND TERMINATOR
        MOV     DL,[BX+3]               ;DL = 2ND DATA TERMINATOR
        CMP     DL,20H                  ;IS THERE A 2ND TERMINATOR?
        JNE     ASR060                  ;BRANCH IF NO 2ND TERMINATOR
;
;       END OF DATA CHARACTERS FOUND // RETURN
;
ASR040:
        MOV     [BX],SI                 ;SETUP INPUT CHARACTER COUNT
        MOV     AL,'Y'                  ;AL = N = SUCCESSFUL COMPLETION
        MOV     AH,'0'                  ;AH = 0 = SUCCESSFUL COMPLETION
        JMP     ASS020                  ;BRANCH TO RETURN TO CALLER
;
;       CHECK FOR 2ND DATA TERMINATOR
;
ASR050:
        MOV     DL,[BX+3]               ;DL = 2ND DATA TERMINATOR
        CMP     DL,AL                   ;IS THIS 2ND TERMINATOR?
        JE      ASR040                  ;BRANCH IF 2ND DATA TERMINATOR
;
        SUB     CX,CX                   ;CX = ZERO = 1ST TERMINATOR
        JMP     ASR030                  ;BRANCH TO CHECK 1ST TERMINATOR
;
        PAGE
;
;       DECREMENT INPUT BUFFER COUNT // END OF BUFFER?
;
ASR060:
        MOV     DX,[BX]                 ;DX = REMAINING BUFFER COUNT
        DEC     DX                      ;DECREMENT BUFFER COUNT
        MOV     [BX],DX                 ;RESTORE REMAINING BUFFER COUNT
        CMP     DX,0                    ;IS THERE REMAINING DATA?
        JNE     ASR010                  ;BRANCH TO READ NEXT CHARACTER
;
        MOV     [BX],SI                 ;SETUP BUFFER COUNT
        MOV     AH,'9'                  ;AH = 9 = END-OF-BUFFER
        MOV     AL,'Y'                  ;AL = Y = SUCCESSFUL COMPLETION
        MOV     DL,[BX+2]               ;DL = 1ST DATA TERMINATOR
        CMP     DL,20H                  ;IS THERE A 1ST TERMINATOR?
        JE      ASR070                  ;BRANCH IF A 1ST TERMINATOR
;
        MOV     AL,'N'                  ;AL = N = ERROR COMPLETION
;
;       SETUP END-OF-BUFFER STATUS // AH = 9
;
ASR070:
        JMP     ASS020                  ;BRANCH TO RETURN TO CALLER
;
ASPORT  ENDP
;
SUBSEG  ENDS
        END     ASPORT                  ;REAL TIME CONCEPTS, INC
```

PC pande
Call us for

SOFTWARE

ApTec *(for color Prism Printers)*
Rainbow Writer Color Text Formatter . . . $119.
Rainbow Writer Screen Grabber . . . . 69.
Ashton Tate
dBase II (DOS or CP/M-86) . . . . 397.
Best Programs
PC/Personal Finance Program . . . . 65.
PC/Fixed Asset System . . . . 239.
Bible Research
THE WORD *(KJV Bible — 7 disks)* . . . . . 149.
Continental
Home Accountant Plus . . . . 95.
FCM *(was First Class Mail)* . . . . 79.
Conceptual Instruments
The Desk Organizer *(30 day return guarantee if not satisfied)* . . . . 250.
Digital Research
CP/M-86 . . . . 39.
DR LOGO . . . . 109.
Financier
Financier Personal Series . . . . 119.
Financier Tax Series . . . . 119.
Hayden
PIE:Writer . . . . 149.
Hayes
Smartcom II . . . . 79.
IUS *(now XT and DOS 2.0 compatible)*
EasyWriter II . . . . 199.
EasySpeller II *for EasyWriter II* . . . . 129.
EasySpeller 1.1 *for EasyWriter 1.1* . . . . 83.
Accounts Receivable . . . . 339.
Accounts Payable . . . . 339.
General Ledger . . . . 339.
Package Price for above three . . . . 879.
Inventory . . . . 339.
Order Entry . . . . 339.
Lifetree
Volkswriter . . . . 119.
Volkswriter International *(fully supports 5 languages)* . . . . 149.
Link Systems
Datafax . . . . 189.
Lotus Development
1-2-3 *(version 1A — the newest!)* . . . . call
Micropro
WordStar 3.3 . . . . 269.
ProPak *(WordStar/MailMerge/SpellStar/StarIndex)* . . . . 397.
Microsoft
Multiplan . . . . 169.
Multi-Tool Word *(with mouse)* . . . . call
PCsoftware
PCrayon *(create in full color with easy to use graphics tool)* . . . . 39.
CREATABASE . . . . 59.
Peachtree
PeachPak (G/L, A/R, A/P) *see above special*
PeachText 5000 . . . . 237.
Pearlsoft
Personal Pearl . . . . 189.

PC Connection Software Special
through September 30, 1983

PEACHTREE
PEACHPAK (Series 4)
The most professional accounting system available for under $300
- Includes A/R, A/P, and G/L
- A/R and A/P will stand alone or update G/L
- Runs under DOS 1.1
- Includes samples and demo data disks
- Detailed manuals make it easy to use

Complete PeachPak System . . . . **$ 237.**

Peter Norton
Norton Utilities . . . . $55.
Rosesoft
ProKey *(customize your keyboard)* . . . . . . 59.
Software Arts
TK!Solver . . . . 219.
Financial Management Pack . . . . 87.
Mechanical Engineering Pack . . . . 87.
Software Publishing
PFS/File . . . . 97.
PFS/Graph . . . . 97.
PFS/Write . . . . 97.
PFS/Report . . . . 87.
Softword Systems
Multimate (was Wordmate) . . . . 329.
Sorcim
Supercalc II . . . . 159.
System Software Services
PCModem 1.3 *(for Smartmodem 300)* . . . . 39.
PCModem 1.4C *(for Smartmodem 1200)* 69.
VisiCorp
VisiCalc 256k (w/free CDEX training program — *limited time)* . . . . 165.
VisiFile . . . . 199.
VisiSchedule . . . . 199.
VisiTrend/Plot . . . . 199.
VisiWord (w/free VisiSpell mail in — *limited time)* . . . . 269.

EDUCATIONAL

Comprehensive
PC Pal *(for first time users)* . . . . 27.
PC Tutor *(more in depth instruction)* . . . . . . 55.
FriendlySoft
FriendlyWare/PC Introductory Set . . . . 39.
Individual Software
The INSTRUCTOR . . . . 35.
Spinnaker
Educational Games for Young Computer Users (require graphics board)
Snooper Troops I *(ages 10 to adult)* . . . . . . 30.
Snooper Troops II *(ages 10 to adult)* . . . . . 30.
Most Amazing Thing *(ages 10 to adult)* . . . 27.
Face Maker *(ages 4 to 12)* . . . . 23.
Story Machine *(ages 5 to 9)* . . . . 23.
Kinder Comp *(ages 3 to 8)* . . . . $23.
Rhymes and Riddles *(ages 5 to 9)* . . . . . . 20.
Hey Diddle Diddle *(ages 3 to 10)* . . . . 20.
Stone *(requires graphics board)*
My Letters, Numbers, and Words *(great graphics and sound — ages l to 5)* . . . . . . 29.

GAMES

Broderbund
Apple Panic . . . . 23.
EPYX/Automated Simulations
Temple of Apshai . . . . 27.
Upper Reaches of Apshai . . . . 15.
Curse of Ra . . . . 15.
Star Warrior . . . . 27.
Rescue at Rigel . . . . 23.
Crush, Crumble, and Chomp . . . . 23.
FriendlySoft
FriendlyWare/PC Arcade . . . . 39.
Funtastic/
Snack Attack II *(a favorite)* . . . . 27.
Cosmic Crusader *(as good as above)* . . . . 27.
Big Top *(hours of fun!)* . . . . 29.
Master Miner *(1 or 2 players)* . . . . 29.
Infocom
High quality text games
Zork I . . . . 27.
Zork II . . . . 27.
Zork III . . . . 27.
Starcross . . . . 27.
Deadline . . . . 35.
Suspended . . . . 35.
Witness *(NEW)* . . . . 35.
Microsoft
Flight Simulator *(with 93 page manual)* . . . 35.
ScreenPlay *(mono or graphics board)*
Asylum . . . . 29.
Dunzhin *(with voice)* . . . . 29.
Sierra/On-Line
Frogger . . . . 27.
Crossfire *(keyboard or joystick)* . . . . 23.
Ulysses and the Golden Fleece . . . . 27.
Strategic Simulations
The Warp Factor . . . . 27.
Tigers in the Snow . . . . 27.
Epidemic . . . . 25.
Sublogic
Night Mission Pinball . . . . 29.
Zee Programming
Zuran Defender *(highly addictive)* . . . . 25.

HARDWARE

AST Research
(For IBM-PC or XT)
We are a full line AST Research Dealer
All AST Boards come with SuperDrive, SuperSpool, and one year warranty.
SixPakPlus 64k *upgradable to 384k, with clock calendar, serial and parallel ports (game port optional)* . . . . *see above special*

TECH NOTEBOOK 4

Tracing a Bug in the 8088

IBM's 8087 comes packaged with a new 8088. Here's one reason why.

WILL FASTIE

According to the Intel 8088 Handbook, a MOV to segment register instruction and a POP segment register instruction are treated similarly: no interrupt is recognized until after the *following* instruction. This is a very important feature of the processor because it allows both parts of an address (segment and offset) to be dealt with as a unit. This is particularly critical when the address of the stack is being manipulated. For example, a program

Open the cabinet and examine the processor chip.

that is changing to a new stack (such as the task switcher in a real-time system) usually updates both SS and SP. If an interrupt were to be recognized after SS had been changed, but before SP has been altered, the processor would push the CS and IP flags into the wrong area of memory.

A BUG!

Certain 8088 parts, including many installed in IBM Personal Computers, *allow* an interrupt during the MOV SS, MOV SP sequence. Intel has since repaired the defect and issued a new chip which operates according to their handbook's specification. The newer part was made available sometime during 1982, so a good number of earlier PC s contain the bad 8088.

There are two ways to tell which version of the 8088 a particular PC contains. The first is to open the cabinet and examine the chip. The 8088's are manufactured with the copyright date printed on the surface of the chip. The older part bears a copyright date of 1978, while the newer one displays a "78,81" date.

TRACING IT!

The second method involves the use of the DOS debugger and shows the bug in action.

Enter the debugger by issuing the command DEBUG from the DOS prompt. Use the R command to check the contents of the AX and IP registers. The value for AX should be 0 and the value for IP should be 100. If they are different, use the R command to change them.

In DOS 2.0 it is possible to enter instructions mnemonically. Give the assemble command as:

```
-A 100
```

to which DEBUG will reply:

```
XXXX:0100
```

where "XXXX" is the segment address which varies from configuration to configuration. Type the following three instructions (DEBUG prompts shown below for clarity):

```
XXXX:0100 MOV ES,AX
XXXX:0102 INC AX
XXXX:0103 INC AX
XXXX:0104
```

Complete the entry with a final ENTER key depression.

For DOS 1.0 or 1.1, use the enter instruction E to enter the hexadecimal value corresponding to the three instructions shown above. The session in this case is:

```
-E 100
XXXX:0100 xx:8E xx.C0
   xx.40 xx.40
```

Once the program has been entered, give the command T. This command "traces" the next *single* instruction: it allows the CPU to execute the next instruction, and then stops execution and returns to the debugger. A dump of all registers is displayed.

Look at the AX register value. If it is 0, the 8088 is the older version. If 1, the newer chip is installed.

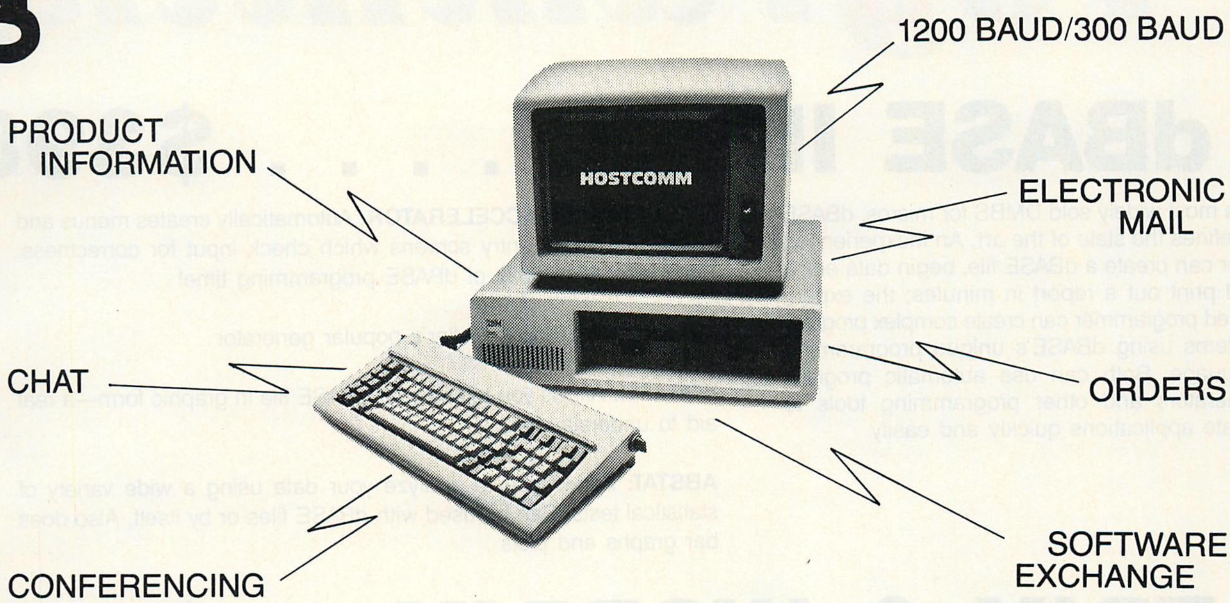
What does this mean? If the value is 0, the INC AX instruction following the MOV was not executed, contrary to the Intel specification. If the value is 1, the instruction *was* executed as expected.

A PROBLEM FOR YOU?

It is certainly not reassuring to discover that the CPU in your computer is not quite right. Whether or not it poses a problem is hard to determine.

It appears that the 8087 may require the newer chip for reliable operation if only because IBM has included a new 8088 in their numeric processor offering. If your application involves the use of any multitasking program, the new chip may help. And finally, if your system has experienced any unexplained "hang-ups" or "stuns," the new 8088 might eliminate them.

Best bet: install the new processor to be worry free. ■



HOSTCOMM

by JANADON **$170.00**

The Complete Host Communications System

★Features:

- The first Host communications system for the IBM Personal Computer
- Run programs from a remote site with a terminal or another computer
- Use as a personal communications system or electronic mailbox
- Full Upload/Download facility for ASCII files
- Easy set-up, installation, and especially easy to use
- Full system operator support
- Three level password protection
- Unattended operation

ADD-ON FEATURES for HOSTCOMM

PCE-MAIL — designed to allow Hostcomm users to leave messages and correspondence for other users. Passwords permit privacy, "wildcard" addresses permit generalized mail. Help files provided. **$80.00**

CONFERENCING — make your Host the most popular system in town. Choose up to 12 different topics for your users. One menu can be used to control both PCE-MAIL and CONFERENCING. **$80.00**

HOST UTILITIES I — a series of 4 utilities for use with HOSTCOMM. MENU-MOD simplifies menu management. Other options compute file sizes and transmit times, list menus and filespecs. **$49.95**

HOST UTILITIES II — HOSTMENU permits alteration of a Download Menu, or changes to a higher number menu. HOSTFILE uses a password scheme so that only certain callers may download files. **$49.95**

ORDER TRANS — a program used with HOSTCOMM to provide callers with information about your products. Product descriptions are contained in a separate file for easy modification. Pertinent order information may be requested and stored from callers. **$39.95**

★Minimum Systems Requirements:

- 128KB System Unit
- 320KB Diskette storage (1 or 2 drives)
- PC-DOS 1.0 or 1.1
- Printer
- Asynchronous Communications Adapter
- Hayes Stack™ Smartmodem 300 or 1200
- Video Display (40 or 80 characters)

IBM®
Personal Computer

NF SYSTEMS, LTD.
P.O. Box 76363, Atlanta, GA 30358

404-252-4146 — DATA 404-252-3302 — VOICE
Add $2.50 for postage and handling

VISA/MASTERCARD

Catalog Available

Programming

dBASE II™ $398

The most widely sold DMBS for micros, dBASE II defines the state of the art. An inexperienced user can create a dBASE file, begin data entry, and print out a report in minutes; the experienced programmer can create complex program systems using dBASE's unique programming language. Both can use automatic program generators and other programming tools to create applications quickly and easily.

THE DATABASE ACCELERATOR: Automatically creates menus and "intelligent" data entry screens which check input for correctness. Eliminates 80—100% of dBASE programming time! **$195**

QUICKCODE: Fox & Geller's popular generator. **$185**

dGRAPH: Allows you to see your dBASE file in graphic form—a real aid to understanding. **$195**

ABSTAT: Allows you to analyze your data using a wide variety of statistical tests. Can be used with dBASE files or by itself. Also does bar graphs and plots. **$359**

EDIX & WORDIX . . $279

EDIX is a full-screen editor specifically designed to take advantage of the advanced features of the IBM PC; WORDIX is the best text formatter on the market. Together, they are an unbeatable combination, the finest word processor available on a micro—at an unbeatable price!

EDIX alone **$149**

WORDIX **$149**

MULTIPLAN™ $189

The world's hottest-selling spreadsheet, Multiplan™ is now setting the standard for ease of use and quality of documentation. Each of the Multiplan™ series comes complete with sophisticated training programs.

Multi-Tool Budget 'expert system': used with Multiplan, enables new users to get budgets out in minutes. **$69**

Multi-Tool Financial Statement: Expert help in getting out financials. **$109**

SMARTMODEM 300 $209

1200 $509

1200B $479

These Hayes Stack Modems link your computer's RS232 serial port directly to a modular phone jack. They will dial the phone for you, and answer it as well. The Smartmodem 300 runs at 0—300 baud; for those needing higher speed, the Smartmodem 1200 can run at 1200 baud.

NEW! NEW! NEW! **Smartmodem 1200B:** for the IBM PC only........... **$469**

(800) 222-8811

International

IBM PC SOFTWARE (PC/MS DOS)

Business

	LIST	SALE
American Training International		
Power for PC DOS	75	56
Aptek		
Rainbow Writer		125
Ashton-Tate		
Financial Planner	700	489
Aspen Software		
Grammatik	75	56
Proofreader 32K	50	38
Proofreader 50K	50	38
Proofreader 80K	50	38
BPI Systems Inc.		
Personal Accountant	195	160
Beaman Porter		
Powertext	399	369
Best Products		
Personal Financial Pgm.	95	66
Comprehensive Software Support		
PC Tutor	80	60
Comshare		
Financial Modeling	325	264
Continental Software		
FCM (First Class Mail)	125	79
Home Accountant Plus	150	99
Datamost		
Real Estate Inves. Prg.	130	95
Eagle Software Publishing		
Money Decisions Vol. 1	199	129
Money Decisions Vol. 2	230	169
Money Pack Vol. 1 & 2	400	299
Emerging Technology		
Edix	195	149
Edix + Wordix	390	279
Howard Software		
Real Estate Analyzer	250	185
Information Unlimited Software		
AP	595	389
AR	595	389
Easy Planner	250	188
Easy Speller 1.1 (Dead)		
Easy Speller 2	225	135
Easy Speller 2/Legal	350	229
Easy Speller 2/Medical	350	229
Easy Writer II	350	229
Financial Mgmt. (AR/AP/GL)	1495	1019
GL	595	389
INV	595	389
Order Entry	595	389
Payroll	595	389
Intellect Associates		
PC Text	100	73
Window	150	113
Lexisoft		
Spellbinder	495	259
Lifetree		
Volkswriter	195	129
Metasoft		
Benchmark Mail List	250	184
Benchmark Word Processor	500	367
Micro Pro		
Mailmerge	250	165
Spellstar	250	165
Word/Mail	695	426
Word/Spell	695	426
Word/Spell/Mail	845	558
Wordstar	495	327
Microsoft ®		
Multiplan™	275	189
Oasis		
Punctuation & Style	0	109
The Word Plus	150	112
PBL Corporation		
Personal Investor	145	98
SERIES 4 PAK (GL,AR,AP)	595	349
Pro-Tem		
Notebook	150	
Scripps Data Systems Inc.		
Job Cost System	495	371
Select Information Systems		
Select Word Processor	595	339
Softward		
Wordmate	500	340
Software Arts		
TK!Solver		224
Software Products International		
Logicalc	190	142
Procalc	350	262
Sorcim		
Supercalc	195	129
Supercalc 2	295	185
Superspellguard	195	129
Superwriter	295	179
Supersoft		
Optimizer	200	149
Texasoft		
Thinker (Spreadsheet)	75	56
Versatext	200	172
Visicorp		
Business Forecasting Modl	100	78
Desktop Plan	300	229
Visicalc	250	169
Visispell	225	183
Visiwork	375	298

Database

	LIST	SALE
Alpha Software		
Database Manager	245	169
Mail List	95	67
Applied Software Technology		
QBASE	189	143
Versaform	389	259
Ashton-Tate		
dBASE II	700	398
dBASE II w/user's guide	729	429
Conceptual Instruments		
Desk Organizer		245
Condor Computer Corporation		
Condor 1	295	189
Condor 3	650	398
Fox & Geller		
Quickcode	295	185
dGRAPH	295	195
dUTIL	99	59
Information Unlimited Software		
Easy Filer	400	279
Innovative Software		
TIM III	495	279
Link Systems		
Data Fax	299	224
Micro Pro		
Infostar	495	327
Reportstar	350	231
North American Business Systems		
<<Answer>>	250	159
PC Software		
Creatabase	75	67
Software Products International		
Logiquest II	250	188
Logiquest III	550	412
Software Publishers		
PFS: File	140	105
PFS: Report	125	94
Supersoft		
Personal Data Base	125	93
Synapse Video		
File Manager	150	112
Synergistic Software		
Data Reporter	250	169
Visicorp		
Visidex	250	189
Visifile	300	219

Hardware

	LIST	SALE
Hayes Microcomputer Products		
Smartmodem 1200B	599	479
Microsoft ®		
Mouse™	195	149

Home/Education/Games

	LIST	SALE
Acorn Software		
Lost Colony	30	22
Lost Colony	30	22
Aptek		
PC Ticker Tape		22
Automated Simulations/Epyx		
Curse of Ra	20	15
Temple of Apshai	40	30
Upper Reaches of Apshai	20	15
Avalon Hill		
Andromeda Conquest	23	17
Computer Stocks & Bonds	25	19
Draw Poker	21	16
Galaxy	25	19
Midway Campaign	21	16
Voyager	25	19
Bible Research		
THE WORD Processor		146
Blue Chip		
Millionaire	100	75
Broderbund Software		
Apple Panic	30	22
Datamost		
Pig Pen	30	22
Space Strike	30	22
Friendly Software		
Friendlyware	50	38
Infocom		
Deadline	50	37
Starcross	40	30
Suspended	50	38
Zork I	40	30
Zork II	40	30
Zork III	40	30
Information Solutions, Inc.		
25:01 The 25th Hour	100	74
ISM		
Mathemagic	90	65
Lightning		
Master Type	50	38
Microsoft ®		
Flight Simulator™	50	38
Omric		
Blingsplats	35	26
Championship Draughts	35	26
Space Guardian	30	22
Sierra On Line		
Crossfire	30	22
Frogger	35	26
Ulysses & Golden Fleece	40	29
Sirius		
Conquest (Call To Arms)	30	22
Strategic Simulations		
Warp Factor	40	30

Language

	LIST	SALE
Howard Sams		
Programmer	200	150
Lifeboat		
C-Food Smorgasborg		
Lattice C Compiler	500	415
Microsoft ®		
C Compiler™	500	400
Mumath/Musimp™	300	225
Pascal™	500	400
muLisp/MuStar™	250	188
Ryan-McFarland		
RM/Cobol Full Dev System	950	713
RM/Cobol Runtime Only	250	188

™ = Manufacturer's trademark

Utility/Comm/Graphics

	LIST	SALE
Alpha Software		
Apple-IBM Connection	195	135
Typefaces	125	87
Aptek		
Epson to Prism		38
Byrom Software		
BSTAM	200	149
BSTMS	200	149
Cavalier		
Championship Blackjack	40	30
Central Point		
Copy II PC	40	35
Hayes		
Smartcom 2	119	90
Hercules		
Graphics CARD	499	399
Innovative Software		
Fast Graphs	295	159
Insoft		
Data Design	225	169
ISM		
Graphmagic	90	65
Graphmagic Combo	150	119
Microstuf		
Crosstalk/Smartmodem	195	135
Peter Norton		
Utilities	80	60
Southeastern		
Data Capture	120	90
Versa Computing		
Graphics Hardcopy	25	19
Versawriter Graphics Tabl	299	249
Visicorp		
Visitrend/Plot	300	215
Woolf		
Move It	150	99

IBM PC HARDWARE

Hardware

	LIST	SALE
AST Research		
Combo + 64K	595	429
MEGA + 64K	595	429
Corona		
10 MB Hard Disk	2295	
5MB Hard Disk	1795	
Corvus		
PC Interface, Cable, Man.	300	239
Kraft		
Joystick	70	52
Microsoft ®		
128K RAM™	525	379
192K RAM Card™	700	499
256 RAM Card™	875	624
64K RAM™	350	249
64K RAM Chip Set™	175	139
Programming International		
PC-Hayes Cable	35	29
QCS		
Big Blue	595	449
Quadrum		
256K QUADBOARD	595	435
64K Quadboard	395	289
64K RAM Chip Set	95	75
Asynch Adaptor, RS232	110	90
Quadcolor I	295	220
Symtec		
Light Pen	150	126
TG Products		
Joystick	65	49
Trackball	65	49
Tandon		
TM100-2 Drive (5¼" DS)	650	249

PC SOFTWARE — CP/M-86

Business

	LIST	SALE
Comshare		
Financial Modeling	325	264
Dictronics		
Random House Thesaurus	150	113
Metasoft		
Benchmark Mail List	250	184
Benchmark Word Processor	500	367
Structured Systems		
AP	1250	790
AR	1250	790
GL	1250	790
INV	1250	790
Order Entry	1250	790
PR	1250	790

Database

	LIST	SALE
Ashton-Tate		
dBASE II	700	398
Fox & Geller		
Quickcode	295	195
dGRAPH	295	195
dUTIL	99	59
Organic		
Datebook	395	331

Language

	LIST	SALE
Digital Research		
CB86 (avail soon!)		
CBASIC 86	200	150
CIS COBOL 86	800	600
CP/M-86	60	45
Concurrent CP/M-86	350	264
Level 2 COBOL 86	1600	1200
Pascal MT+ 86	400	300
Pascal MT+86 With SPP-86	600	450
SID 86	150	94
Micro Focus		
CIS Cobol 8086	850	637
Ryan-McFarland		
RM/Cobol Full Dev System	950	713
RM/Cobol Runtime Only	250	188

Utility/Comm/Graphics

	LIST	SALE
Byrom		
BSTAM	200	149
Woolf		
Move It	150	99

HARDWARE, NOT MACHINE SPECIFIC

	LIST	SALE
Amdek		
300 (12" green)	175	142
Color 11 (R&G)	899	645
Corona Data Systems		
11MB Hard Disk	3295	2695
20MB Hard Disk	4295	3495
6MB Hard Disk	2295	1895
PC Interface,Cable,Man.	300	239
Corvus Systems		
11MB Hard Disk	3295	2695
20MB Hard Disk	4295	3495
6MB Hard Disk	2295	1895
Hayes Microcomputer Products		
Chronograph	249	199
Smartmodem 1200	699	509
Smartmodem 300	279	209
NEC		
Monitor 12" Green Hi-Res	285	167
Novation		
J Cat	149	119
Smart Cat 103	259	207
Smart Cat 103/212	595	476
Quadrum Corp.		
Microfazer	189	160
Microfazer, in-line serial	330	270
Microfazer inline paralle	225	185

® = Microsoft is a registered trademark

Orders must be **PAID** by **October 31** to qualify for these special prices

TELEPHONE ORDERS

Outside Calif.:	(800) 222-8811
Inside Calif.:	(415) 324-3730

Order lines are manned 6:30—6 Monday thru Friday and 9—5 Saturday. Other lines are open 9—5 Monday thru Friday.

Technical Support	**(415) 324-0311**
Order Status	**(415) 324-0306**
Sales Manager	**(415) 324-0305**

More CP/M®, APPLE®, IBM PC™, UCSD p-System™ software, hardware, etc.: call for quote.

TERMS: All prices subject to change without notice and availability. Cashier's check/MO/bank transfer. Allow time for company or personal checks to clear. Prices reflect cash prepaid discount. VISA/MASTERCARD/COD/PO's + 3%. CA residents add sales tax. All sales final for games & special orders.

SHIPPING: $3 per item for UPS surface ($6 for Blue Label) within Continental USA, except where shipping cost is specified in square brackets. UPS does not go to Canada, Alaska, APO's, FPO's; call for ship charge or add 15%—we will refund/credit difference.

RETURNS: Must have authorization number, obtained at 415-324-0305. Unauthorized returns will be refused; damaged goods will be refused. All returns subject to 15% restocking fee. No return after 30 days.

PROGRAMMING INTERNATIONAL

505 Hamilton Avenue Suite 301
Palo Alto, California 94301

Also available at

Software to go

Call (800) 222-8811, (415) 324-1099, or local information for Software to go location nearest you. Software to go prices may differ slightly. Local sales tax applies.

OPENING SOON

California
- Los Angeles
- Orange County
- San Diego
- Palo Alto

Illinois
- Chicago

New Jersey
- Camden
- North Jersey

New York
- Manhattan

Texas
- Dallas-Ft. Worth
- Houston

to serve you better!

CIRCLE NO. 250 ON READER SERVICE CARD

PROGRAM EDITORS

18 Program editors offer a wide variety of choices. Here are our picks and pans.

SUSAN GLINERT-COLE

This is an article about text editors. It is not an article about word processors per se, but concerns itself specifically with those pieces of software that are best suited for writing computer programs, as opposed to lengthly manuscripts. One can view a text editor as a primitive word processor that lacks internal formatting capabilities. A program file must, of necessity, be composed only of standard ASCII characters; compilers will not recognize embedded formatting commands and will halt abruptly when these types of character strings are encountered.

While their most expedient use is generally considered writing program source code files, text editors can also function as minimal word processors. The formatting is done by arranging the text on the display exactly as desired for hard copy, because editors function in the "what you see is what you get" mode. Should the user wish to make the output look fancier than may be conveniently done with the editor, the file may be processed with a text formatter program. Many of the editors reviewed in this article come with some formatting capabilities, while others have companion formatters that can be purchased separately. Because text editors are designed to permit only standard ASCII characters in the file, a favorite formatter and spelling checker may be combined with the text editor of choice to produce a complete word processing package that has the best of three worlds.

The excellent editors had meaningful prompts, excellent error handling, intuitive cursor movement and useful help screens.

As with most things in life, all text editors are not created equal, more is not necessarily better, and, in general, you get what you pay for. On the whole, the more expensive editors have an extensive battery of commands that allow some very complex and elaborate text maneuvers. Some of these facilities are so complicated that it seems unlikely that they would ever be used on a regular basis. On the other hand, the cheapest editors were very limited and usually had abysmal documentation. There was no one editor that was a clear winner; the best editor is the one that best suits your requirements, your personal taste in implementations, and, of course, your pocketbook. Some features important to one person may well be superfluous, or even irritating, to another.

BEST BUYS, BAD BUYS

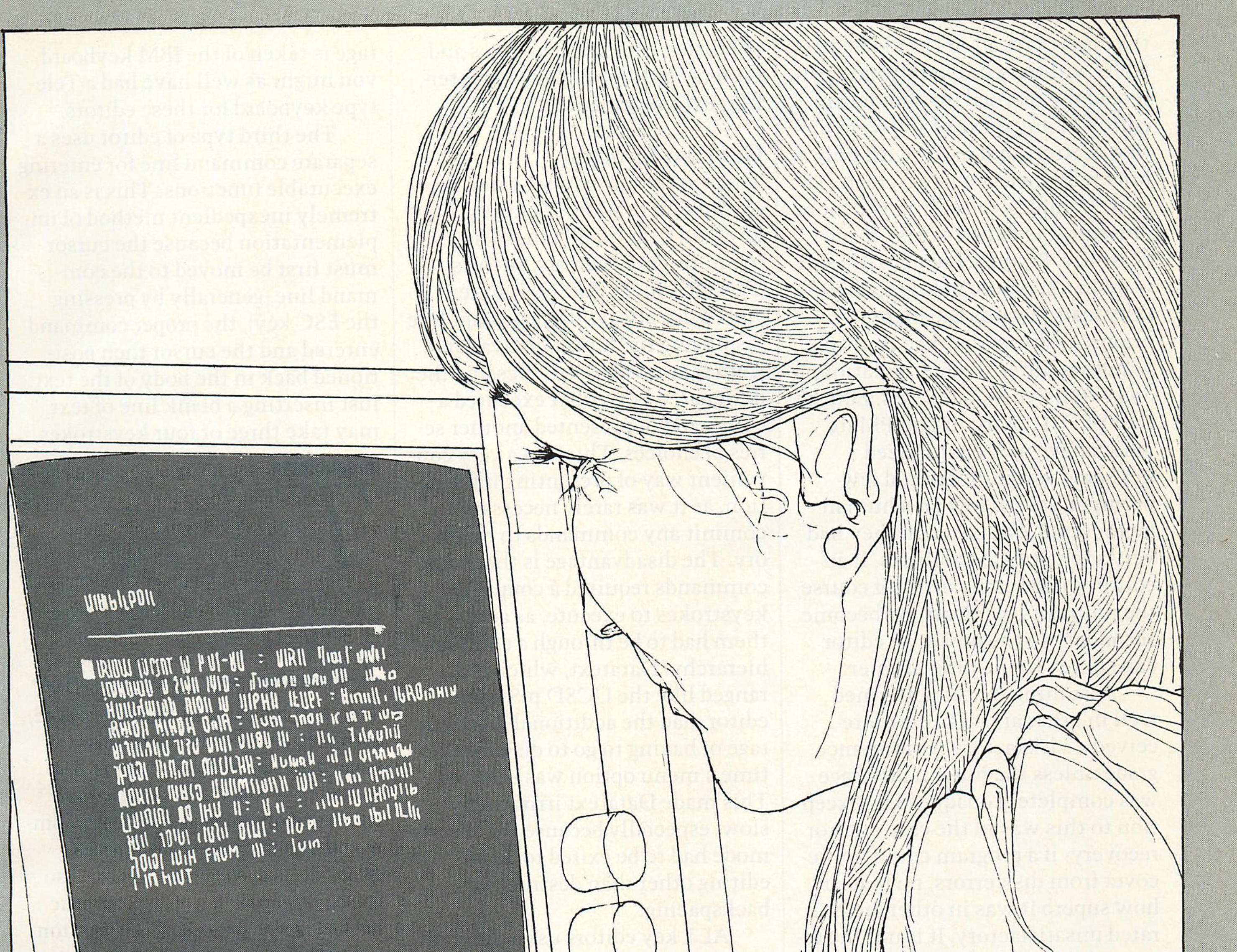
The editors ranged in quality from outstanding to downright user-surly. The best buys, listed in table 2, incorporated the features I felt to be the most important ones for easy program entry; they all implemented these functions in a consistent and pleasant manner. There were some outstanding bad buys too; these are listed in table 3. The worst editors lost files when the keyboard was accidentally bumped, crashed inexplicably when an incorrect key was pressed and sometimes reacted unpredictably to documented commands. There is really very little excuse for this type of software, no matter how inexpensive it may be.

FACTORS HUMAN

Many reviews of this type often list the functions available in each piece of software without giving any indication of the quality of the human interface. Some of the editors had an amazing range of capabilities, but were so difficult or awkward to use that they were almost worthless. The excellent editors had meaningful prompts, excellent error handling, intuitive cursor movement, and useful help screens. These factors combine to make a package that, if not used for a period of time, gracefully bears

Susan Glinert-Cole recently joined the TECH JOURNAL *staff as a technical editor. She also writes the monthly column "IBM Images" in* Creative Computing.

(Tables 2 and 3 appear on page 123.)



the onus of recall. It should be possible, without completely having to reread the manual, to boot up an editor (or any piece of software for that matter), and use it with only a brief reference to a help screen. The more complex the editor, the more difficult it is to carry this off successfully. Because I tend to use software in a sporadic manner, I naturally lean towards the menu-oriented editors that had multiple windows, interfile manipulation, and extensive help screens. Editors like VEDIT, StarEdit and PMate, which have non-customized keyboard assignments and few prompts, scored low on a human factor scale, even though they had very sophisticated abilities. If an editor is used constantly, of course, the problems of relearning become secondary to the power an editor can supply to the programmer.

Assuming that it performed well in other areas, an editor received a satisfactory performance grade unless the human interface was completely opaque. The exception to this was in the case of error recovery; if a program could not recover from disk errors, no matter how superb it was in other areas, it rated unsatisfactory. It is not the responsibility of the user to keep track of disk space, and losing an entire file because the disk is full is simply unacceptable.

At the very least, a text editor should be able to 1) permit text entry in both insert and overstrike

If a program could not recover from disk errors, no matter how superb it was in other areas, it rated unsatisfactory.

mode, 2) move copy and delete blocks of text, 3) do string search and replacement, and 4) have impeccable file handling routines. These functions are sufficient to allow efficient program entry. The more lavish editors offer macro capabilities, multiple windows and buffers, and comprehensive interfile text movement.

Getting Started

There are three distinct categories of implementation: Menu, ALT key, and Command Line. The menu-oriented editors, such as TEXTRA Jr., SCREEN, Datatext and I-Edit, always had a prompt line displaying the functions available from that menu level. Pressing the designated key either executed a command or presented another series of choices. This was a very convenient way of presenting information, as it was rarely necessary to commit any commands to memory. The disadvantage is that some commands required a couple of keystrokes to execute, as access to them had to be through a command hierarchy. Datatext, which is arranged like the UCSD p-System editor, had the additional disadvantage of having to go to disk every time a menu option was selected. This made Datatext irritatingly slow, especially because the Insert mode had to be exited to do any editing other than destructive backspacing.

ALT key editors use a different approach. All the commands, such as MOVE, COPY, and DELETE, are evoked by pressing ALT (or CTRL) plus some letter key. The creators of EDIX did an excellent job of making these assignments as mnemonic as possible, so they were very easy to remember. P-Edit's commands are not quite so intuitive, but it comes with keyboard templates for the absentminded. Other editors, such as VEDIT, PMATE, and Star-Edit, used more mystical methods of key assignments, and constant reference to the manual is required to recall the proper command sequences. While these editors are very powerful, they are not configured specifically for the IBM PC and require considerable practice to use effectively. The manuals contain an insert that rather grudgingly gives out a few key assignments, but, in general, no advantage is taken of the IBM keyboard; you might as well have had a Teletype keyboard for these editors.

The third type of editor uses a separate command line for entering executable functions. This is an extremely inexpedient method of implementation because the cursor must first be moved to the command line (generally by pressing the ESC key), the proper command entered and the cursor then positioned back in the body of the text. Just inserting a blank line of text may take three or four keystrokes.

The IBM professional editor managed to combine the worst of all possible worlds.

This can become a laborious chore, especially if you are the cautious type who saves the file every five minutes or so. Many people like this kind of editor because the command can be issued in plain English: delete, copy, change, and so forth. XyWrite was definitely the best in this category. The function keys were assigned in a very convenient manner, the program was forgiving as to the exact format of the command, the cursor keys were intelligently assigned, and once one got used to its style of use, it was a pleasure to use.

The IBM Professional Editor managed to combine the worst of all possible worlds. It is extremely complex, or, to be more exact, overly complicated. There are line commands as well as screen commands, four operating modes, 40 assignments for the function keys, and one of the busiest displays of all the editors. Even with the powerful capabilities that the IBM version possesses, it and other command line editors do not seem to offer any particular advantage that outweighs the inconvenience of using them and they tend (with the exception of Xywrite) to be very awkward to use.

If you've been hearing some great lines lately about buying computers, software and accessories by mail, wait until you hear ours.

The bottom one.

You've probably heard them all — "We won't be undersold," "We offer competitive prices," on and on. But when it comes to **consistently** low prices on **consistently** high quality computers, software and accessories, no one can show a better bottom line.

But a good bottom line doesn't mean anything if you're not getting the kind of service you deserve. The kind you can expect from the seasoned group of mail order professionals at Oryx Systems. We also have the technical know-how to stand behind the products we carry — before **and** after the sale.

The following is a sampling of prices on selected items from our extensive product line of over **3,000 items:**

SOFTWARE

Ashton-Tate: dBase II — $399 / **Digital Research:** CP/M86 — $50 / Pascal MT+ W/SPP — $375 / **Ecosoft:** MicroStat — $230 / **Dow Jones:** Market Analyzer — $279 / **Innovative Software:** T.I.M. III — $365 / **MicroPro:** InfoStar — $275 / 3-Pak Word/Mail/Spell — $399 / **Northwest Analytical:** StatPak — $397 / **Sorcim:** Supercalc 2 — $169 / **Supersoft:** Fortran IV PC/DOS or CP/M86 — $299 / 8087 Support — $40 / **Visicorp:** Visicalc — $185 / **Compview:** Vedit — $130 / **Decision Resources:** Chartmaster — $310 / **Lotus:** 1-2-3 — $ CALL / **Microsoft:** Multiplan — $199 / Multitool — $ CALL / **Peachtree:** PeachText 5000 — $247 (includes one free box of diskettes) / Ms Pascal Compiler — $275 / Business Basic Compiler — $275

ACCESSORIES / HARDWARE

BOARDS

Quadram: Quadboard 64K (4-function) — $299 / Microfazer 8K — $145 / **Plantronics:** Color Plus — $399 / **Maynard Electronics:** Sandstar Memory Card OK — $194

MONITORS

Amdek: 310A — $189 / **Quadram:** Quadchrome — $ CALL / PGS RGB — $ CALL

MODEMS

Hayes: Smartmodem 1200 — $509 / **US Robotics:** Passport — $395

DISK DRIVES

Tandon TM-100-2 — $240 / **CDC:** 1800 — $275

PRINTERS

NEC: 3550 — $1,875 / **Transtar:** T-315P (color) — $519

MISCELLANEOUS

MicroSoft: Mouse — $179 / Versa Writer Tablet — $259

If you're ready to order, call our toll free number. Right now. (Chances are we'll ship the same day.) Tell us what you're thinking about **buying**, and we'll tell you our bottom line. Simple as that. It won't cost you anything. In fact, it might **save** you plenty — and that's not just another line.

Wisconsin residents — add 5% sales tax when ordering merchandise. Add $3.50 for shipping per software and small items. (Call regarding others.) Prices subject to change without notice. All items subject to availability. We welcome VISA and Mastercard (Add 4%), checks (allow 1-2 weeks for clearing), and COD (Add $2.00 per shipment — cashier's check required). For technical information and in Wisconsin: (715) 848-2322. Store prices differ from mail order.

1 800 826-1589

ORYX SYSTEMS, INC.

205 Scott Street
P.O. Box 1961
Wausau, WI 54401

TABLE 1: PROGRAM COMPARISON

EDITOR	AFSE	DATA-TEXT (6.1)	EDIT	EDIT SMITH (1.0)	EDIX	FULL SCREEN EDITOR	IBM PROF. EDITOR	I-EDIT (2.05)	P-EDIT	PMATE (3.22)	SCREEN (2.0)	STAR-EDIT	TED (1.0)	TEXTRA JR.	VEDIT (1.15)	XY-WRITE	BIG ED	EASY EDIT
REQUIREMENTS																		
MINIMUM MEMORY	64K	96K CPM/86	48K	64K	128K	?	96K	48K	48K	?	96K		128K	64K	64K	48K	128K	48K
RECOMMENDED MEMORY	128K	128K MS-DOS			192K	?	128K			?	128K							
DISK DRIVES	1	1	1	1	1	1	1	1	1	?	1	2	2	1	2	1	1	1
OPERATING SYSTEM	MS-DOS	MS-DOS	MS-DOS	MS-DOS	MS-DOS	MS-DOS	MS-DOS	MS-DOS	MS-DOS	MS-DOS	MS-DOS	MS-DOS (CPM-86)	MS-DOS	MS-DOS	MS-DOS (CPM-86)	MS-DOS	MS-DOS	MS-DOS
OTHER		CPM-86	IBM MO. DISPLAY														IBM MO. DISPLAY	
COPIES ALLOWED	YES	YES		YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES
DOCUMENTATION																		
PAGES	61	106	11	63	130	7	271	15	50	79	49	120	58	54	152	80	25	NOT
FORMAT	SOFT COVER	LARGE BINDER	STAPLED	SOFT COVER	IBM BINDER	STAPLED	IBM BINDER	STAPLED	IBM BINDER	STAPLED	IBM BINDER	LARGE BINDER	SOFT COVER	IBM BINDER	LARGE BINDER	IBM BINDER	SOFT COVER	AVAIL-
TABLE OF CONTENTS	YES	YES	NO	YES	YES	NO	YES	YES	YES	NO	YES	YES	NO	YES	YES	YES	NO	ABLE
INDEX	NO	YES	NO	NO	NO	NO	YES	NO	YES	NO	NO	NO	YES	YES	YES	NO	NO	AT
REFERENCE CARD	NO	YES	NO	NO	NO	NO	YES	NO	NO	NO	NO	YES	YES	NO	NO	NO	NO	REVIEW
OVERALL QUALITY	GOOD	EXCEL.	FAIR	POOR	EXCEL.	POOR	GOOD	FAIR	GOOD	POOR	GOOD		POOR	EXCEL.	GOOD	EXCEL.	GOOD	TIME
TUTORIAL																		
ON SCREEN	NO	NO	NO	NO	YES	NO	NO	NO	YES	NO	NO		NO	YES	NO	NO	NO	NOT
IN MANUAL	YES	YES	NO	YES	NO	NO	YES	NO	YES	YES	NO	YES	YES	YES	YES	YES	NO	AVAIL-
TIME REQUIRED	½ HOUR	2 HOURS	–	1 HOUR	½ HOUR	–	½ HOUR	–	1 HOUR	1 HOUR	–		½ HOUR	2 HOURS	½ HOUR	½ HOUR	–	ABLE
OVERALL QUALITY	FAIR	EXCEL.	–	POOR	EXCEL.	–	FAIR	–	FAIR	FAIR	–		POOR	EXCEL.	FAIR	GOOD	–	
MAXIMUM FILE SIZE	20,000 CH. (64K) 40,000 CH. (128K)	MEDIA BOUND	MEDIA BOUND	64K CHAR.	40,000 CH. (128K)	61,000 CHAR.	64K CHAR.	48,000 CHAR.	64K	MEDIA BOUND	50,000 CHAR.	MEDIA BOUND	61,500 (128K) 300,000 (256K)	14K (64K) 60K (128K)	MEDIA BOUND	MEDIA BOUND	32,000 CHAR.	?
MAXIMUM LINE LENGTH	254 CHAR.	255 CHAR.	NONE	255 CHAR.	254 CHAR.	72 CHAR.	140 CHAR.	80 CHAR.	NONE	250 CHAR.	240 CHAR.		75 CHAR.	NONE	260 CHAR.	80 CHAR.		80 CHAR.
WORD WRAP OPTION	YES	YES	NO	YES	YES	NO	YES	YES	NO	YES	NO	YES	NO	YES	YES	YES	YES	YES
AUTO INDENT	NO	YES	NO	NO	NO	NO	YES	NO	NO	YES	NO	YES	NO	YES	YES	NO	NO	YES
AUTO LINE NUMBERING	NO	NO		NO	NO	YES	NO	NO	NO	NO	NO		YES	NO	NO	NO		NO
TABS																		
SET TABS	NO	YES	YES	YES	YES	YES	YES	YES	YES	YES	NO	YES	YES	YES	YES	NO	YES	YES
STORED AS BLANKS	YES	YES	YES	YES	NO	YE	YES	YES	YES	YES	YES	NO	YES	YES	YES	NO	YES	YES
STORED AS TAB CHARS	YES	NO	YES	NO	YES	NO	YES	NO	YES	YES	NO	YES	NO	NO	YES	YES	NO	YES
HALF TABS	NO	NO	YES	NO	NO	NO	NO	YES	YES	NO	NO		NO	NO	NO	NO	NO	
SET MARGINS	NO	YES	NO	YES	YES	NO	YES	YES	NO	YES	NO	YES	NO	YES	YES	YES	NO	YES
SET PAGE SIZE	NO	NO	NO	NO	NO	YES	YES	NO	NO	YES	NO	NO	YES	NO	NO	NO	YES	NO
CURSOR MOVEMENT																		
LEFT/RIGHT BY WORD	NO	YES	YES	YES	YES	NO	YES	YES	YES	YES	YES	YES	NO	YES	YES	YES	YES	YES
UP/DOWN A LINE	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	NO	YES
UP/DOWN N LINES	NO	YES	NO	NO	NO	NO	YES	NO	YES	YES	NO	YES	NO	NO	YES	NO	NO	YES
BEG/END OF LINE	YES	NO	END ONLY	YES	YES	NO	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES
BEG/END OF BUFFER	–	NO	NO	YES	YES	NO	–	NO	–	NP	–	YES	–	–	–	–	–	YES
BEG/END OF SCREEN	NO	NO	NO	YES	YES	NO	NO	BEG. ONLY	YES	NO	YES	YES	NO	YES	YES	YES	NO	YES
BEG/END OF PAGE	–	NO	NO	–	–	–	–	–	YES	–	–	–	–	–	–	–	–	–
BEG/END OF FILE	NO	YES	YES		YES	YES	YES	YES	YES	YES	YES	YES	NO	YES	YES		NO	YES
JUMP TO LABELS	NO	YES	NO	YES	NO	NO	NO	NO	YES	YES	YES		NO	YES	YES	NO	NO	NO
JUMP TO A LINE NUMBER	YES	NO	NO	NO	YES	YES	NO	NO	NO	NO	NO		YES	NO	NO	NO	NO	YES
JUMP TO A PAGE NUMBER	NO	NO	NO	NO	NO	NO	NO	NO	YES	NO	YES		NO	NO	NO	NO	NO	YES
PGUP/PGDN	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES
INTUITIVENESS	FAIR	FAIR	FAIR	EXCEL.	EXCEL.	POOR	FAIR	POOR	EXCEL.	POOR	EXCEL.		FAIR	GOOD	POOR	EXCEL.	EXCEL.	EXCEL.
TEXT DELETION																		
BY CHAR	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES
BY WORD	NO	YES	NO	YES	NO	NO	NO	YES	YES	YES	YES	YES	NO	YES	YES	NO	YES	
BY LINE	NO	YES	NO	YES	YES	YES	YES	NO	NO	YES	NO	YES	YES	YES	YES	YES	NO	YES
TO END OF LINE	YES	NO	NO	YES	NO	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	NO	YES	YES
TO END OF SCREEN	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO		NO	YES	NO	NO	NO	NO
TO END OF PAGE	NO	NO	NO	NO	NO	NO	NO	NO	YES	NO	NO		–	–	–	–	NO	–
RANGE SELECTABLE	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES
'UNDO' KEY	NO	YES	YES	YES	YES	NO	YES	YES	NO	YES	NO	YES	NO	YES	YES	YES	YES	NO

EDITOR	AFSE	DATA-TEXT (6.1)	EDIT	EDIT SMITH (1.0)	EDIX	FULL SCREEN EDITOR	IBM PROF. EDITOR	I-EDIT (2.05)	P-EDIT	PMATE (3.22)	SCREEN (2.0)	STAR-EDIT	TED (1.0)	TEXTRA JR.	VEDIT (1.15)	XY-WRITE	BIG ED	EASY EDIT
TEXT MOVEMENT																		
BLOCK MOVE	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES
BLOCK COPY	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES
BLOCK HIGHLIGHTED	NO	NO	YES	YES	YES	NO	NO	YES	NO	NO	YES		NO	NO	YES	YES		NO
SWAP CHARACTERS	NO	NO	NO	NO	NO	NO	NO	YES	NO	NO	NO	YES	NO	NO	NO	NO	NO	NO
SWAP WORDS	NO	NO	NO	NO	NO	NO	NO	YES	NO	NO	NO		NO	NO	NO	NO	NO	NO
SWAP LINES	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	YES		NO	NO	NO	NO	NO	NO
SEARCH AND REPLACE																		
FORWARD	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES
BACKWARD	NO	YES	YES	YES	YES	YES	YES	NO	YES	YES	YES		NO	YES	YES	YES	NO	YES
GLOBAL REPLACE	YES	YES	NO	YES	YES	NO	YES	YES	YES	YES	YES		YES	YES	YES	YES	YES	YES
LOCAL REPLACE	YES	YES	NO	YES	YES	NO	YES	NO	YES	YES	YES		YES	YES	YES	YES	YES	YES
REPEAT LAST S/R	NO	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES		YES	YES	YES	YES	NO	YES
CASE CHANGE OPTION	NO	NO	NO	NO	NO	NO	NO	NO	YES	NO	NO		NO	NO	YES	NO	NO	NO
CASE IGNORE OPTION	NO	YES	NO	NO	NO	NO	NO	NO	NO	NO	YES		NO	YES	YES	NO	NO	NO
WILDCARDS	NO	NO	NO	NO	YES	NO	NO	NO	NO	YES	NO		NO	NO	YES	NO	NO	NO
ALIGNMENT																		
SHIFT LINE	NO	YES	NO	NO	NO	NO	NO	NO	NO	NO	NO		NO	YES	NO	NO	NO	YES
SHIFT PAGE	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO		NO	YES	NO	NO	NO	YES
CENTER LINE	NO	YES	NO	NO	NO	NO	YES	NO	NO	NO	YES		NO	YES	NO	NO	NO	YES
CENTER TEXT	NO	NO	NO	NO	NO	NO	YES	NO	NO	NO	NO		NO	NO	NO	NO	NO	YES
CASE CONVERT	NO	NO	NO	NO	NO	NO	YES	YES	YES	NO	YES	YES	NO	NO	NO	YES	NO	NO
FILE EDITING																		
MULTIPLE WINDOWS	NO	NO	NO	YES	YES	NO	NO	NO	NO	NO	NO	YES	NO	NO	NO	NO	NO	YES
MULTIPLE BUFFERS	NO	NO	NO	YES	YES	NO	NO	NO	NO	YES	NO	YES	NO	NO	YES	NO	NO	YES
MULTIPLE FILE EDITING	NO	NO	NO	YES	YES	NO	NO	NO	NO	NO	NO		NO	NO	YES	NO	NO	YES
INTERFILE MOVE/COPY	NO	YES	NO	YES	YES	NO	NO	NO	NO	YES	NO		NO	YES	YES	NO	NO	YES
BLOCK MOVE TO A FILE	NO	NO	YES	YES	YES	YES	YES	NO	YES	YES	YES		NO	NO	YES	YES	YES	YES
FILE MERGING	NO	YES	YES	YES	YES	NO	YES	YES	YES	YES	YES		YES	YES	YES	YES	YES	YES
FILE APPEND ONLY	–	–	–	–	–	NO	–	–	–	–	–		–	–	–	–	–	–
SPACE AVAIL. PROMPT	YES	YES	NO	NO	YES	YES	YES	YES	NO	NO	YES		NO	YES	NO	NO		NO
DISK I/O																		
READ DIRECTORIES	NO	YES	YES	NO	YES	NO	YES	NO	YES	YES	YES		NO	NO	YES	YES	NO	NO
ERASE FILES	NO	YES	YES	NO	YES	NO	NO	NO	YES	YES	YES		NO	NO	YES	YES	NO	NO
RENAME FILES	YES	YES	YES	NO	YES	NO	NO	NO	NO	NO	NO		NO	NO	YES	NO	NO	NO
FILE SIZE	YES	YES		NO	YES	NO	YES	YES	NO	NO	NO		NO	NO	NO	YES		NO
FULL DISK RECOVERY	YES	YES		NO	YES	YES	YES	NO	YES	NO	YES		YES	YES	YES	YES	NO	YES
DISK ERROR RECOVERY	YES	YES		YES	YES	YES	YES	NO	YES	NO	YES		YES	YES	YES	YES	NO	YES
FILE SAVE																		
WITH BACKUP	NO	YES			YES	YES	YES	YES	YES	YES	YES		YES	YES	YES	NO	YES	YES
WITHOUT BACKUP	YES	NO		YES	NO	NO	YES	NO	YES	NO	NO		YES	YES	NO	YES	NO	NO
AUTOSAVE	NO	NO		NO	NO	NO	NO	NO	NO	NO	NO		NO	NO	NO	NO	NO	NO
EXIT CONFIRM	NO	YES	YES	NO	YES	NO	YES	YES	YES	NO	YES		YES	YES	YES	NO	YES	NO
SAVE AND RETURN	YES	YES		NO	YES	NO	YES	YES	YES	YES	YES		YES	NO	YES	YES	YES	YES
PRINT FACILITIES																		
PRTSC ONLY	NO	–	YES	YES	–	–	–	–	NO	–	YES		NO	YES	–	–	YES	NO
FANCIER	YES	YES	NO	NO	YES	YES	YES	YES	YES	YES	NO		YES	NO	YES	YES	NO	YES
PRINT SPOOLING	NO	YES	YES	NO	NO	NO	NO	NO	NO	NO	NO		NO	NO	NO	YES	NO	NO
FORMATTER AVAILABLE	YES (INCLUDED)	YES		YES (EXTRA)	YES (EXTRA)	NO	NO	YES (INCLUDED)	NO	YES (INCLUDED)	NO	NO	NO	NO	YES (INCLUDED)	YES	YES	
MACROS	NO	NO		YES	YES	NO	YES	NO	YES	YES	NO		NO	NO	YES	NO	NO	YES
HUMAN FACTORS																		
HELP KEY	NO	NO	YES	YES	YES	NO	NO	NO	YES	NO	YES		YES	YES	NO	YES	YES	YES
QUALITY OF HELP	–	–		FAIR	EXCEL.	–	–	–	FAIR	–	GOOD		POOR	EXCEL.	–	GOOD		GOOD
HUMAN INTERFACE	FAIR	GOOD		GOOD	EXCEL.	POOR	FAIR	FAIR	EXCEL.	POOR	GOOD		POOR	EXCEL.	POOR	EXCEL.		FAIR
PRICE	$29.95	$179	$35	$75	$125	$29.95	$130	$35.00	$95	$195	$75	$225	$95	$39.95	$195	$50	$125	$49.95

PROGRAM EDITOR SUMMARY

	AFSE	DATA-TEXT (6.1)	EDIT	EDIT SMITH (1.0)	EDIX	FULL SCREEN EDITOR	IBM PROF. EDITOR	I-EDIT (2.05)	P-EDIT
SUMMARY									
GENERAL TYPE	COMMAND LINE	MENU	COMMAND LINE	ALT. KEY	ALT. KEY	COMMAND LINE	COMMAND LINE	MENU	ALT. KEY
EASE OF USE	GOOD	GOOD		GOOD	EXCELLENT	POOR	FAIR	GOOD	EXCELLENT
ERROR HANDLING	FAIR	GOOD		FAIR	EXCELLENT	FAIR	FAIR	POOR	EXCELLENT
OVERALL PERFORMANCE	SATISFACTORY	SATISFACTORY		SATISFACTORY	SATISFACTORY	UNSATISFACTORY	SATISFACTORY	UNSATISFACTORY	SATISFACTORY
OVERALL RATING	FAIR	GOOD		FAIR	EXCELLENT	POOR	FAIR	POOR	EXCELLENT
COMMENTS	VERY SLOW	BASED ON P-SYS. EDITOR SLOW-MUCH DISK I/O	NO REPLACE FUNCTION	AWFUL DOCUMENTATION		IRRITATING SCREEN FLICKER EXTREMELY RUDE	COMPLEX	CONTAINS A DICTIONARY NO DISK ERROR RECOVERY	
MANUFACTURER	NOTA BENE SOFTWARE	DATATEK	McCLAIN SOFTWARE	SYMMETRIC SOFTWARE	EMERGING TECH.	FRANK SHIELE SOFT.	IBM CORP.	INNOVATIVE TECH. INC.	SATELLITE SOFT. INTL.
ADDRESS	BOX 497 GLENVIEW, IL 60025	2621 ENTERPRISE ROAD CLEARWATER, FL 33575	P.O. BOX 383 TURNERSVILLE, NJ 08012	1805 CLEMSON ST. SAN BERNADINO, CA 92407	2031 BROADWAY BOULDER, CO 80302	1375 TOBIAS DRIVE CHULA VISTA, CA 92011	P.O. BOX 1328-C BOCA RATON, FL 33432	2387 BLACKSTONE CRES. OTTAWA, ONTARIO	288 W. CENTER ST. OREM, UT 84057
PHONE	(312) 724-9442	(800) 237-5014	(609) 338-4631 (DAYS) (609) 228-6461 (EVES)	(714) 887-8595	(303) 447-9495	(619) 420-9860		(613) 521-8722	(801) 224-8554

PMATE (3.22)	SCREEN (2.0)	STAR-EDIT	TED (1.0)	TEXTRA JR.	VEDIT (1.5)	XY-WRITE	BIG ED	EASY EDIT
ALT. KEY	MENU	ALT. KEY	COMMAND LINE	MENU	ALT. KEY	COMMAND LINE	COMMAND LINE	ALT. KEY
POOR	EXCELLENT		POOR	EXCELLENT	FAIR	EXCELLENT		GOOD
POOR	FAIR		GOOD	GOOD	EXCELLENT	GOOD	POOR	FAIR
UNSATISFACTORY	SATISFACTORY		UNSATISFACTORY	SATISFACTORY	SATISFACTORY	SATISFACTORY	UNSATISFACTORY	SATISFACTORY
POOR	GOOD		POOR	EXCELLENT	FAIR	EXCELLENT		GOOD
CRASHES EASILY	EXQUISITELY PAINFUL NOISES, OTHERWISE GOOD BUY		MANY FUNCTIONS DID NOT PERFORM AS DESCRIBED, IF AT ALL CRASHED IF TABS WERE NOT SET PRIOR TO USE	BEST BUY PRINT PROG. AVAILABLE: FREE UPGRADABLE TO TEXTRA SR. (A WORD PROCESSOR)	COMPLEX BUT VERY POWERFUL	BEST BUY	NO I/O RECOVERY	EDIT FILE MUST BE SPECIFIED ON START-UP. PRELIMINARY VERSION REVIEWED HERE
LIFEBOAT ASSOCIATES	BASIC BUSINESS SOFTWARE	SUPERSOFT INC.	MORGAN COMPUTING CO.	ANN ARBOR SOFT. ASSOC.	COMPUVIEW	XYQUEST	IROHNIC SYSTEMS	NORELL DATA SYSTEMS
1651 THIRD AVENUE NEW YORK, NY 10018	P.O. BOX 26311 LAS VEGAS, NV 89126	P.O. BOX 1628 CHAMPAIGN, IL 61820	10400 N. CENTRAL EXPWY. SUITE 210 DALLAS, TX 75231	407 N. MAIN STREET ANN ARBOR, MI 48104	1955 PAULINE BLVD. SUITE 200 ANN ARBOR, MI 48103	P.O. BOX 372 BEDFORD, MA 01730	11195 25 MILE ROAD UTICA, MI 48087	3400 WILSHIRE BLVD. P.O. BOX 70127 LOS ANGELES, CA
(212) 860-0300	(702) 876-9493	(217) 359-2112	(214) 739-5895	(313) 769-9088	(313) 996-1299	(617) 275-4439	(313) 781-4587	(213) 257-2026

Although almost all of the editors (TED being the exception) made use of the function keys, only two, P-EDIT and SCREEN, thought to provide templates. This is a generous but unnecessary fallal for SCREEN, because it presents the available choices on the prompt line anyway. With the more taciturn editors such as VEDIT and

The documentation varied from elegant, superbly written manuals with outstanding on-screen tutorials (TEXTRA Jr. and EDIX), to seven half sheets of poorly xeroxed paper stapled meanly in the upper left hand corner.

StarEdit, the lack of templates forces the user to commit to memory copious command sequences that are hopelessly un-mnemonic.

Manual Labor

The documentation varied from elegant, superbly written manuals with outstanding on-screen tutorials (TEXTRA Jr. and EDIX), to seven half sheets of poorly Xeroxed paper stapled meanly in the upper left hand corner. Most of the companies obviously put some effort into making the documentation informative and readable. It was, however, appalling to see how many misspelled words most of them contained. The seven page uninformative screed mentioned above that accompanied the Full Screen Editor consistantly misspelled ALT as ATL. This misspelling could have been easily fixed if the author had thought to provide a REPLACE facility. EDITSMITH, however, had the most deplorable manual of them all. The spelling and grammatical errors it contained were incredible (and often hysterically funny). The pity of this is that EDITSMITH is a pretty good editor; unfortunately some of the commands were difficult to figure out from the text. It is a credit to the designer, and his/her respect for the end-user, that the editor can be used without much reference to the documentation

Only a few of the editors had on-screen tutorials. TEXTRA Jr. provides an unusual, hands-off tutorial that is positively elegant. Having gone through the presentation once, it is rarely necessary to refer to the equally well-done manual. EDIX has an excellent hands-on tutorial that gives a fine introduction to the editor but is not as complete as TEXTRA Jr's. P-Edit's on-screen tutorial was, alas, mediocre. It is a bit more difficult to present a text-level tutorial, because switching back and forth from the manual to the display can be distracting. Datatext and Xywrite both managed to do a good job anyway.

What Size the File

The maximum size of an edit file varied widely, but the most important factor here is whether the editor requires the entire file to be resident in memory or whether it can swap segments of the file to and from disk during the editing process, in effect allowing documents of arbitrary size. File sizes for the memory-bound editors are necessarily smaller than those that are media bound, while the media-bound editors will become slower as the edit files become large. None of the files I edited was voluminous enough to produce an obvious effect; all the editors but two were very fast.

As for program performance, AFSE, which is written in BASIC, was so slow that I could out-type it in the INSERT mode. Datatext was very curiously designed to swap each menu function from the disk into memory when the command is invoked. This produced an almost constant disk activity that quickly became annoying. It also took forever to load; I was beginning to suspect a disk malfunction; then, finally, the title page appeared. Considering that almost none of the commands is resident, one could well ask what it was loading during the initialization process.

Editor as Word Processor

Most of the programs were flexible enough to use as primitive word processors, but two of them were oriented strictly towards assembly or BASIC programmers: Full Screen Editor and TED. These two editors had automatic line numbering, which could be a convenience under some circumstances. Coincidentally, both editors also rated unsatisfactory; TED has many functions that could not be performed as described in the manual, and the Full Screen Editor, with its execrable manual, completely defeated my attempts to get line numbers back after inadvertently typing a carriage return in the wrong place. The latter editor also had an intensely irritating screen flicker during text entry and manipulation. It was easy to tell which editors had been written for programmers using high-level languages: these had an automatic indentation feature. Writing programs in PASCAL and C without an auto-indent feature can quickly degenerate into a tedious exercise on the tab key. Vedit also had an UNDENT key for rapid outdenting. The other editors with auto-indent (IBM, TEXTRA, Jr, DataText, EasyEdit and PMate) re-

EDITSMITH, however, had the most deplorable manual of them all. The spelling and grammatical errors it contained were incredible (and often hysterically funny).

How to get a fast 300% return on your IBM-PC investment.

Compared with COBOL interpreters, mbp's new COBOL Compiler runs programs four times faster. With mbp on your IBM-PC, applications that used to take four hours now take one; that's like a 300% return on investment.

How is it possible?

Interpreters translate and execute a program one statement at a time, every time the program runs. mbp's COBOL Compiler generates machine language object code, so the entire program is translated only once; it then can be executed as often as you want in a fast single step—with no retranslation.

Gibson Mix: BENCHMARK RESULTS
Calculated S-Profile
(Representative COBOL statement mix)
Execution time ratio

mbp COBOL	COBOL X	COBOL Y
1.00	4.85	7.04

And it's COBOL—

the language for commercial and administrative applications. Standardized and maintained like no other language, now COBOL brings mainframe benefits to business microcomputer users.

mbp COBOL meets ANSI '74 Standard. It features a full range of utilities: 'Compiler Intelligence' automatically corrects simple syntax errors; Run-time Debug; Index Check and Array-Bound Inspection; listing options allow source and object code, map and cross-reference tracking. 500+ error messages insure user-friendliness. There is plenty of existing software, and maximum portability protects software investments.

Just $500.

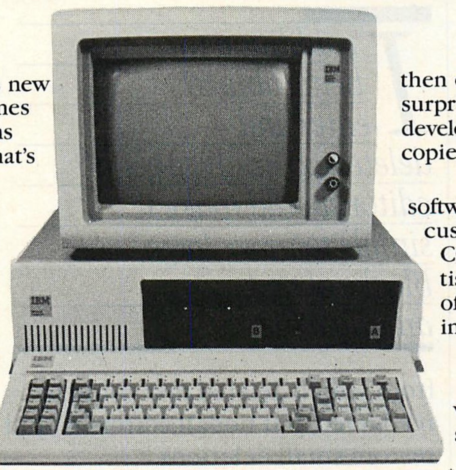
Compare mbp's advantages, then compare prices; you'll be pleasantly surprised— again. Plus, application software developers can make up to 50 run-time package copies with no royalty charge.

mbp has twenty-five years worldwide software experience and twelve(+) in mainframe custom COBOL implementations. The mbp COBOL Compiler makes this valuable expertise available for micro users. Take advantage of it, and start getting a 300% return on investment, today.

mbp COBOL is available now

for the IBM-PC* or any micro computer with MS-DOS,* iRMX-86,* or CP/M-86;* soon for OASIS-16.*

*IBM-PC is an IBM TM; MS-DOS, Microsoft TM; iRMX-86, Intel TM; CP/M-86, Digital Research TM; OASIS-16, Phase One Systems TM.



IBM-PC +

mbp®

7700 Edgewater Drive, Suite 360
Oakland, CA 94621 Phone 415/632-1555

Please send me complete mbp COBOL information and performance data.

Name ______

Company ______

Address ______

City ______

State ______ Zip ______

Phone # ______

OEM & distributor arrangements available.

mbp COBOL.
300% faster–just $500.

CIRCLE NO. 165 ON READER SERVICE CARD

quired multiple backspaces to outdent.

Most of the editors incorporated a word-wrap feature that, when turned on, relieves the user of having to type carriage returns at the end of a line. Word-wrap is a plus when the editor is being used as a word processor, but for programming it is best turned off, as many compilers have an upper limit for program line length.

The ability to set tabs is a very important feature in a program editor; only SCREEN and AFSE did not allow this to be done by the user. Many of the editors also allowed tab settings to be saved in a file, so that several preset values could be recalled for FORTRAN, PASCAL, and assembler programs. In almost every case, tab characters could be expanded to blanks when written to the file; in some cases the choice of storage format could be set by the user. While storing tabs as blanks takes up more space on a disk, some compilers will not accept tab characters and the ability to chose between storage fomats is thus a necessity.

The IBM keyboard lends itself to convenient cursor movement; it was therefore very surprising to find a few editors that did not take advantage of this. In fact, some of them, in an attempt to give the user the widest range of cursor movements, assigned these functions to totally non-intuitive keys. Although intuitiveness is a matter of opinion and habit, it seemed unreasonable to use ESC to go to the beginning of a file or CTRL-V to move to the previous word. The cursor movements that get around a file most easily, such as left/right by word, beginning/end of a line, screen and file, are the ones that are found in the best editors. Some of the command line types, such as EDIT, have very limited cursor movements and getting around a file can be painfully slow.

Paging through a file was almost always accomplished with the PgUp and PgDn keys. In some cases it is possible to set a specific page size for scrolling through a file. This is a pleasant feature that can be used to size a page for printing.

The ease with which text may be deleted is equally as important as easy text entry. I like the convenience of deletion by word, by line, and to the end of a line with one key press. TEXTRA Jr. and P-Edit were the only ones that could delete to the end of a page, a feature I use very frequently. P-Edit gives a prompt before executing the page delete while TEXTRA Jr. does not. Many people prefer to be safe than sorry; others dislike the necessary extra keystroke.

The presence of an "undo key" is a feature I would not like to do without either. Most of the editors that could cut blocks of text for deletion, move, or copy, use a buffer

In addition to being able to enter and delete text easily, a good editor should make it simple to move large blocks of text around a file.

from which deleted text could be recovered, but not all editors would restore small deletions.

Text Moves

In addition to being able to enter and delete text easily, a good editor should make it simple to move large blocks of text around a file. The most convenient implementations required only three or four keystrokes: two to mark the block limits, one to select MOVE, COPY, or DELETE, and one to indicate the insertion place. This was one feature that almost all editors accomplished in a simple manner; many visually highlight the text range as well. This feature makes it very easy to see what you are doing if the editor is accurate about the highlighting (I-Edit isn't). AFSE, for some strange reason, emits a tone when the markers are set, and even supplies a different tone for MOVE/COPY and DELETE. This feature is described as useful in the manual, but I found it hard to recall where the tone was placed in the text body. P-Edit does not highlight the text, but has the added advantage (as does EDIX) of allowing the text range to be defined from start to end or vice versa.

XyWrite, SCREEN, P-EDIT, I-EDIT and IBM had a case convert feature that operated on a marked block of text. If the characters were in lower case, they were changed to upper case and vice versa. This is a clever alternative to toggling the CAPS LOCK key on and off. A very few editors could, with one keystroke, exchange two lines, words, or characters. This is another little cute feature, but may not be used very often, if at all.

Search and replace functions are indispensible in a text editor, even if used only infrequently. Full Screen Editor could only find strings; no replace facility was included. The nicest implementation of these features always had a prompt for both commands, didn't require tiresome string delimiters to be typed in, searched the file in both directions, and allowed wildcards to be included in the strings. EDIX was the only one that could perform to these standards, VEDIT made this relatively painless function impossibly difficult to execute, and the rest fell in the middle somewhere.

If a search direction has to be specified in VEDIT, and an entire file search is requested, the cursor must be moved to the top (for a forward search) or the bottom of the screen (for a backwards search). The function may be case sensitive, in which case it will find only the correct string matches. If it is case insensitive it will locate all occurrences of the string, regardless of its state of capitalization. Datatext, SCREEN, VEDIT, and TEXTRA Jr. will ignore case differences if instructed to do so. VEDIT and P-Edit

have a CASE CHANGE option, so the search can be automatically repeated without retyping the string. All the editors, with the exception of I-Edit, could do both a local or global string replacement. I-Edit will only do a translate throughout the entire file; if the user wishes to do a local or prompted replacement with this editor, it must be done using the search function with manual string insertion.

TEXTRA Jr., Datatext and EasyEdit were the only editors that included line and page alignment. This excellent feature uses the cursor keys to pull or push a line across the screen and align former or subsequent lines to the same mark by running the cursor up and down the display. Not only does this facilitate the neatening of PASCAL's innumerable begin/end clauses, but it makes column alignment a snap. These editors will also (as will IBM and SCREEN) center text lines, which is useful for table headings.

Lavish Functions

Macros are an advanced capability that many people find essential in a text editor. These are strings of commands that ideally can be assigned to one key for execution and saved to disk for eternal recall. The most common use made of macros is to specify default values to be called up during initialization. None of the editors that included this function made it easy to do;

None of the editors that included macros made it easy to do; somehow the documentation writers all got terribly confused about macros.

somehow the documentation writers all got terribly confused about macros. This is a feature I found more work to use than seemed worth the effort. In those editors that set switches for word wrap, margins, and so on, the default values were generally those that would be used most of the time anyway. For the more exotic and adventurous, macros offer a way out of the status quo.

It is often desirable to move a marked block of text to a separate disk file. In itself, this function is extremely useful, but in combination with multiple buffers and windows, it allows the editing of several files simultaneously. The editors that have multiple windows, such as EDIX, EDITSMITH, and EASYEDIT, have many buffers available to facilitate text movement between files. Not everyone finds multiple windows a desirable addition to an editor; I use them a great deal and prefer EDIX's implementation over all of the other editors with this capacity. The newest version of P-Edit allows two documents to be edited simultaneously and includes the ability to move text between the two; only one document can be viewed at a time.

Again, these advanced features are a matter of taste, and, in general, the price of an editor rises in proportion to the number of windows it permits on the display at once. At the very least, however, an editor should be able to insert a file from the disk into the text (file merge). AFSE does not have any file merging features at all, and Full Screen Editor only allows an append of one line (72 characters), which is useless for inserting a previously typed subroutine or other block of text. Multiple windows allow you to preview a file before merging; programmers with good memories may not need this option.

Many editors include good file-handling facilities such as deleting and renaming files and reading directories. Editors varied in their attitudes towards swapping diskettes during an edit session. All of the 'satisfactory' editors will recover from a DISK FULL error, but in some cases the file may only be saved by erasing files from the edit disk. Facilities are provided for reading the directory and selecting the files to erase. While this is not as convenient as being allowed to insert a fresh diskette, at least the file will not be irrevocably lost. Even with no errors pending it is

For the more exotic and adventurous, macros offer a way out of the status quo.

nice to be able to browse through a directory and view the file sizes and creation dates without having to exit to DOS.

Saving files can be done with many different options. There was always a QUIT command, which exited without saving the edit session, and an EXIT command, which wrote the file to disk. Some editors, IBM's for example, allowed a quit with no file save, by pressing one key. IBM chose to assign a function key to this process, so one mistaken finger movement can lose the entire edit session. I consider an 'exit confirm' to be a necessity for a program of this kind, but some people dislike the extra bother of having to press an additional key. P-Edit is pleasantly extra-cautious; it will ask you to confirm an overwrite when saving a file with the same name as that on the disk. Unbelievably, TEXTRA Jr., EDITSMITH, and Full Screen Editor did not allow a file save with a return to the editor so frequent file savers have to restart everything after each save.

The editors were divided in their backup capabilities. Some of them always retained a backup file, while others left it to the user to do the housekeeping. A very few editors (TED, Textra Jr., IBM, P-Edit) allowed the option of backup files to be selected from within the editor. Backup files take up a lot of disk space, and some people prefer to have a backup on a different flop-

Table 2: Best Buys

TYPE	EDITOR	REASON
MENU	TEXTRA Jr.	very easy to use, excellent human interface
ALT KEY	EDIX	extremely powerful, excellent human interface
	P-Edit	very easy to use, excellent human interface
COMMAND LINE	Xywrite	very well designed, easy to use, excellent human interface

Table 3: Worst Buys

EDITOR	REASON
FULL SCREEN EDITOR	rude, poor user interface, baffling text entry
IEDIT	no disk error recovery
TED	many functions did not perform as described
PMate	no disk error recovery
StarEdit	no "DISK FULL" recovery, limited to IBM display
Edit	no "DISK FULL" recovery, limited to IBM display

py anyway. The editors that had excellent error recovery never lost a file no matter what I tried and would write the file out to a fresh diskette if directed to do so.

Into Print

In general, print facilities in these editors were minimal. A few simply used the (SHIFT) PrtSc func-

The editors that had excellent error recovery never lost a file no matter what I tried and would write the file out to a fresh diskette if directed to do so.

tion. This is an unacceptable way of handling some printers, like the NEC 8023A, which responds to this command by printing the entire page on one line unless an inaccessible DIP switch inside the printer has been changed. Also, if any command prompts, menus, or other status information are present on the screen, these get printed along with everything else. TEXTRA Jr. provides a free, short BASIC program that will pretty-print the file, but this is an extra, time-consuming step. EDIX required a temporary renaming of the active file to MS DOS's reserved device name "PRN," executing a write command and then reinstating the original file name. This process could be assigned to a macro, which was a mildly tiresome way of handling this type of situation. In all fairness, EDIX is designed to be used with a companion text formatter and TEXTRA Jr.'s big brother, TEXTRA, will have print facilities available but even so, being able to print out the program so laboriously typed in seems so essential that is is difficult to believe that the designers of these two excellent editors couldn't go one little step further.

Some editors had a print command of some kind; this was by far the most convenient way of getting hard copy. P-Edit gave the choice of a full text, page or range print, which is how this facility ought to be implemented.

Caveat Emptor

At least a day was spent reading the manual and exploring the capabilities of an individual editor. The tables that follow will give a good feel for the ease of use, documentation quality, and sophistication level of each one. Because of the number of editors that were covered in this review, it simply was not possible to explore every single feature of every single piece of software. When the documentation and the program itself combined to give a product that was awkward or functionally inadequate, the summary reflects this problem. Because more time had to be spent trying to figure out how to use the editor due to poor manuals or program design, less time could be allotted to actually using the editor.

The function of a review of this nature is to inform the buyer of the best and the worst cases. Toward that end, this reviewer has tried to be as fair as possible in evaluating these products. Even the best editors had faults that, it is hoped, will be remedied in later versions. As for the unusable editors, well, perhaps this evaluation will simply alert people to stay away from them because they are terribly flawed from both a human factors and program design point of view and cannot be fixed with a few simple patches and a pretty binder. ■

EPSON'S NEW RX-80 AND FX-80

EPSON's two new printers are a world apart.

ARTHUR A. GLECKLER

EPSON recently announced two additions to its line of dot matrix printers: the RX-80, essentially a fast version of the earlier MX-80 III printer, and the FX-80, which has advanced features that put it on top of the printer performance spectrum.

The RX-80's initial retail price of $499 is very interesting indeed, for the MX-80, a lower-quality printer, made its debut with a retail price of $645. This is an example of the trend toward lower prices in the personal computer printer market.

The FX-80, at $699, is not intended to replace the MX-80 as the world's most popular home computer printer; rather, it has been introduced to put EPSON into the race for high-quality, business and scientific printers. The FX-80's download character set and larger printer buffer make it much more suitable for these environments than the MX-80 or RX-80.

Although the MX-80 is no longer being sold to consumers, EPSON is still producing it to fulfill contracts with OEMs (Original Equipment Manufacturers). IBM contracted with EPSON to produce its IBM Graphics Printer, which is almost identical to the MX-80. When IBM's contract with EPSON runs out, it is safe to assume that the RX-80, the FX-80, or both will be added to the PC's product line. Since IBM is just beginning to market a modified version of the PC overseas, it may be particularly attracted to the RX-80's and FX-80's ability to produce foreign language character sets.

GENERAL DESCRIPTIONS

The RX-80, despite its resemblance to the MX-80, has several distinguishing features. The most noticeable difference is its speed; the RX-80 prints at 100 characters per second, while the MX-80 prints at 80 cps. Although not dramatic, the 25 percent improvement does make a difference when printing long documents.

EPSON has also provided new graphics modes for the RX-80. Along with the standard 480-dot and 960-dot print modes, the RX-80 had a 960-dot double-speed mode and high-density, 1920-dot mode.

The FX-80 is truly a superior printer. While the RX-80's increased print speed is a noticeable feature, the FX-80's 160 cps (*double* the speed of the MX-80) is a tremendous improvement. The FX-80 has also been provided with a 2K buffer, which speeds up printing of small documents and reduces the time the computer spends servicing the printer. The FX-80 also has new print functions, including a download character set, proportional spacing, and elite printing.

The FX-80 offers much more versatile paper-handling capabilities than EPSON's other printers. It comes with friction feed and has single sheet feed ability. It also has pin feed, although the tractors can

only be adjusted for widths between 9.5 and 10 inches. This might be limiting in a business setting, where it is often necessary to adjust the distance between tractors to fit mailing labels, envelopes and forms.

EPSON comes to the rescue with a $55 adjustable tractor option, called the FP-80. The tractor mechanism simply snaps onto the existing mechanism, providing fully adjustable tractors (see photo 1).

Physical Characteristics

The RX-80 is very similar in appearance to the MX-80 (see photo 2). It has the roughly the same footprint: 107 mm high × 372 mm wide × 303 mm deep (4.2″ × 14.6″ × 11.9″). The case styling is almost identical to the MX-80's, as can be seen from the fact that its dimensions differ only by millimeters. At 100 mm × 420 mm × 374 mm (3.9″ × 16.5″ × 13.7″), the FX-80 has a larger footprint than the MX-80 or RX-80, although it is slightly shorter.

The FX-80 has a shorter cover than either the MX-80 or RX-80. This redesign allows forms to be removed from the printer more easily. The RX-80 uses the same wire-and-plastic paper separator as the MX-80, and its data cable exits on the left side, just as the MX-80's does. The FX-80 has a plastic paper separator and the data cable exits on the right side.

There are several very interesting innovations in the RX-80 and FX-80. Unlike the previous printers, neither must be turned upside down to open the case. In the RX-80, two screws are located immediately below the printer ribbon cartridge. To open the case, one simply removes the ribbon and screws, disconnects the paper feed knob and buttons, and lifts the cover off on its plastic hinges; certainly much more simple than the MX-80 or IBM Graphics printers.

The FX-80 is even easier to open. Although there are four screws as compared to the RX-80's two, the ribbon need not be removed to open the case. This saves a great deal of time and avoids the mess involved with rethreading the ribbon.

The RX-80 printer has a smaller printed circuit board with fewer integrated circuits than previous EPSON printers. Also, the three 2K-byte ROMs of the earlier printers have been reduced to one 8K-byte ROM.

There are four ROM sockets inside the FX-80. The first two contain 2K bytes of ROM each, but can be replaced with 8K ROMs; the third socket is empty, but can accept a 2K, 8K, or 16K ROM; and the fourth socket holds 8K, but can take 16K. Clearly, the FX-80 has been designed for expansion, both by EPSON and by outside companies.

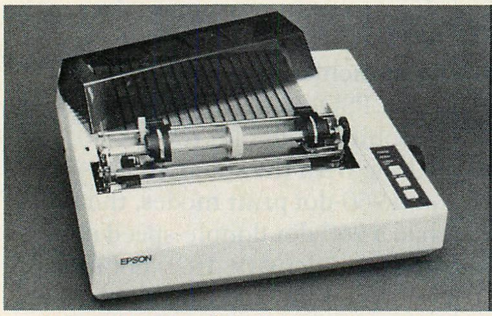
The FX-80 has two expansion board connectors inside; there is provision for only one in the RX-80. On the MX-80, hooking the cable to an expansion board inside the case requires routing the cable through a hole in the middle of the back of the printer. This is very awkward, because the paper enters and exits the printer from the back, and the cable is very much in the way. The RX-80's access hole is in the middle as well. This problem is eliminated on the FX-80 because one of its expansion boards can be located directly above the cable connector on the right side of the case.

By far the most obvious change inside the FX-80 is in the size of the power supply. Both the power transformer and the heat sink for the voltage regulator are much larger than in the MX-80. Obviously, more power is required to drive the printer at twice the speed of the MX-80.

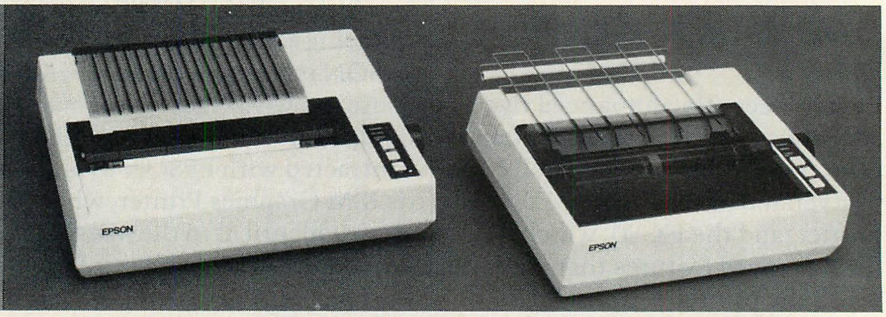
The RX-80's initial retail price of $499 is very interesting indeed, for the MX-80, a lower-quality printer, made its debut with a retail price of $645.

Photo 3 shows the layouts of the insides of all three printers. Note that both the RX-80 and FX-80 use the same ribbons that the MX-80 does; dealers who stock MX-80 ribbons will immediately be able to supply FX-80 and RX-80 users with ribbons.

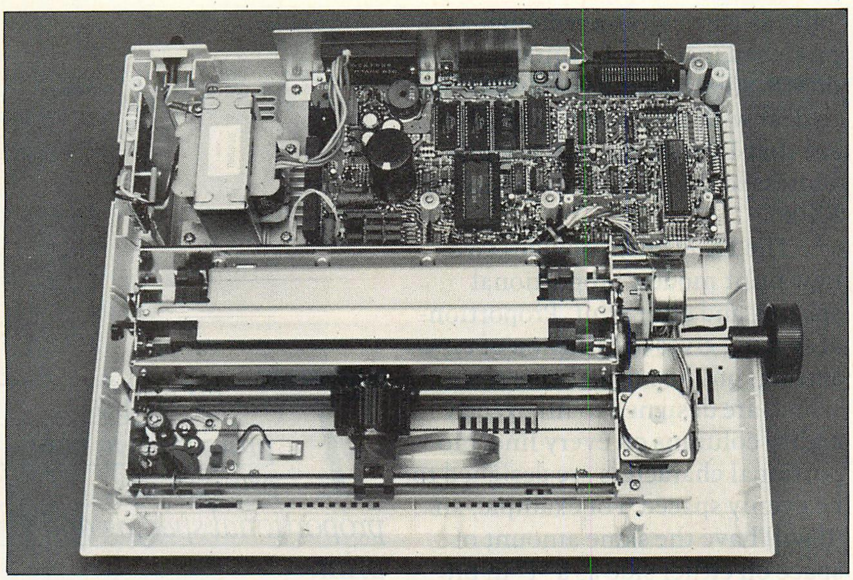
The RX-80 and FX-80 display the same quality of workmanship and materials that have given the MX-80 its legendary reliability. The cases are made of the same strong plastic, the printed circuit boards use the same quality components, and the print mechanism is constructed in the same tough manner,



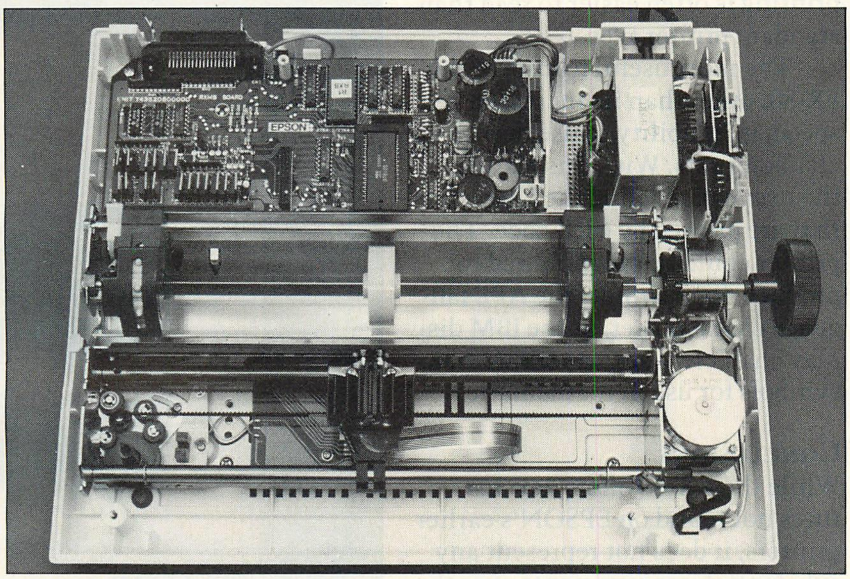
The FX-80 with fully adjustable tractor option



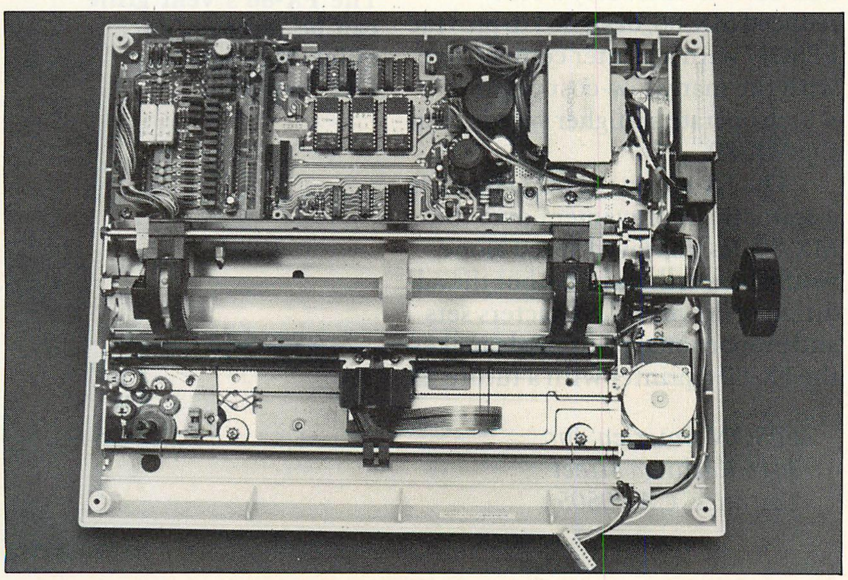
The EPSON FX-80 and RX-80



Inside the printers **FX-80**



RX-80



MX-80 (IBM Graphics)

so it is safe to assume that the RX-80 and FX-80 will prove to be as rugged and reliable as the MX-80 has been.

One problem that has not been corrected with the RX-80 and FX-80: The DIP switches are still not easily accessible. With the RX-80, the printer must be disassembled (i.e. the cover must be removed) to access the switches; while this is easier to do with the RX-80 than the MX-80, for example, it is no small task.

The designers of the FX-80 have taken a small step toward correcting this problem by providing a small door on the side of the printer. With the removal of one screw, the ventilation grille on the right side of the printer can be lifted off, providing access to the switches (see photo 4).

Although few EPSON owners will need to change the default switch settings on their printers, the switches could be placed much more conveniently for those times when the switches simply *must* be changed.

Another interesting difference between the older MX-80 and IBM Graphics printers and the new RX-80 and FX-80 is the operation of the Online button. Although the Online button is used for the same purpose on all four printers (namely, to select and deselect the printer), the newer Online button is much more convenient. On the older printers, the Online button does not work while the printer is printing; it is obviously intended only to deselect the printer while manual paper movement is in progress.

The newer Online button, on the other hand, stops printing at any time and patiently awaits another press. Also, as soon as the printer is brought back online, it prints the contents of its buffer in the correct position, even if printing was stopped in the middle of a line.

Another major improvement in the FX-80 allows the Online button to be pressed at any time; the print-

er remembers the depression and brings the printer online at the next opportunity. For example, if the Online button is pressed in the middle of printing a page, the line currently being printed will be finished and the printer will stop. If the FF button is then pressed, the printer will begin to move to the beginning of the next page. In the newer EPSONs the Online button can be pressed again immediately, and the printer will automatically come back online and print anything that remains in its buffer when the formfeed is completed.

Control Code Function

Although there are no major innovations in the area of print modes on the RX-80, there are some enhancements. One interesting feature is the provision for international characters. The RX-80 provides characters sets for the U.S., France, Germany, the U.K., Denmark, Sweden, Italy, Spain, Japan, and Norway. The FX-80 supports most of the RX-80 international character sets, although it is missing a set for Norway and one of the two Danish sets.

Both printers sport new graphics modes, including a double-speed, 960-dot mode and a high-density, 1920-dot mode.

The FX-80 supports an additional graphics mode: 9-pin graphics. This mode takes advantage of the 9-pin print head of the FX-80. EPSON has done software developers a favor in continuing to support the 8-pin graphics modes even on the 9-pin FX-80, because programs written to handle MX-80 graphics will still work on the newer FX-80. Although 9-pin graphics can be simulated with only 8 pins, 9-pin graphics offer the programmer the ability to take advantage of the speed increase that an additional pin provides.

The FX-80 offers one final feature in the graphics category: it has one-to-one dot spacing. When a program sends an MX-80 the codes which should produce a square (e.g., same number of dots horizontally as vertically), a rectangle is produced instead of a square. Users of MX-80 graphics have always had to deal with this, and it is good to see that EPSON has successfully addressed this problem with in the FX-80.

The FX-80 also provides two new print modes: proportional spacing and elite print. Proportional spacing gives the illusion of typesetting; unlike normal characters, which are designed to fill exactly eighty columns on every line, proportional characters are designed to be evenly spaced. For example, an 'I' will have the same amount of space on either side as a 'T' in proportional print mode. Proportional printing is often easier to read than standard printer type.

The most useful feature of the FX-80, other than its increased speed, is its ability to use download character sets. With this feature, the user can redefine the characters that are printed by each code. The printer could be programmed to produce, for example, the APL language character set, or the IBM display characters set, as well as many symbols for use in mathematics.

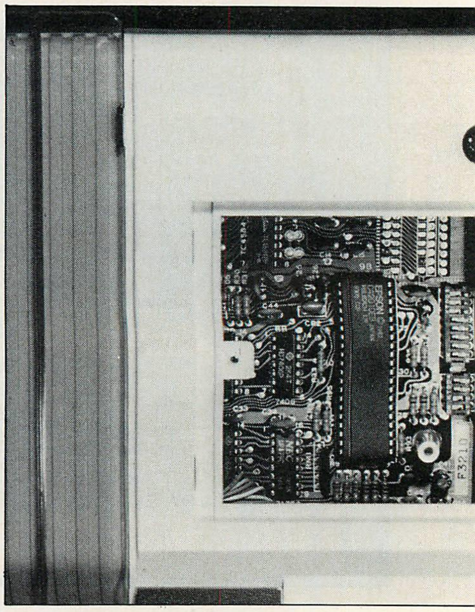
Conclusions

While the RX-80 offers new features not found on EPSON's earlier printers, it does not represent any substantial improvements over the MX-80. It is obviously a cost-reduced MX-80, designed to provide EPSON with a printer competitive with the many low-cost printers which operate at higher speeds than the MX-80.

The FX-80, on the other hand, is a dramatic improvement over the MX-80. Its improved speed alone would be worth its price, but the addition of download characters sets and proportional print mode makes the FX-80 a printer with a future. ■

EPSON America, Inc.
3415 Kashiwa Street
Torrance, CA 90505

The FX-80 also provides two new print modes: proportional spacing and elite print.



The FX-80's vent grille can be removed for quick access to the DIP switches.

The Choice of Professionals™

Computer Exchange — The Supply Center for the IBM-PC or XT

SOFTWARE for the IBM-PC or XT

BUSINESS

		LIST PRICE	OUR PRICE
ASHTON-TATE, dBase II, requires CP M-86 & 128K		$ 700	$439
dBase II, requires PC-DOS & 128K		$ 700	$439
The Financial Planner	NEW!	$ 700	$439
dBase II User's Guide by Software Banc		$ 30	$ 20
Everyman's DB Primer (Book)	NEW!	$ 15	$ 12
Friday	NEW!	$ 295	$199
APPLIED SOFTWARE TECHNOLOGY, Versaform		$ 389	$265
CONTINENTAL, The Home Accountant Plus		$ 150	$112
FCM (Filing, Cataloging, Mailing)		$ 125	$ 85
Property Management		$ 495	$329
DATAMOST, Real Estate Investment Program	NEW!	$ 130	$ 89
Write-On (Word Processor Program)		$ 130	$ 89
DOW JONES, Dow Jones Market Manager		$ 300	$239
EAGLE SOFTWARE, Money Decisions	NEW!	$ 150	$ 109
HAYDEN, IBM Pie Writer	NEW!	$ 200	$135
HOWARDSOFT, Real Estate Analyzer II		$ 250	$189
INFORMATION UNLIMITED, EasyWriter II (Word Processor)		$ 350	$259
EasySpeller II (88K Words)		$ 225	$149
EasyFiler (a DBMS)		$ 400	$299
INNOVATIVE SOFTWARE, T.I.M III (a DBMS)		$ 495	$329
★ INSOFT, Data Design (a powerful easy to use DBMS)	NEW!	$ 225	$169
Data Base Made Easy (Book)	NEW!	$ 17	$ 12
ISM, MatheMagic		$ 90	$ 69
★ LOTUS, 1-2-3	NEW!	$ 495	$329
MICROCRAFT, Verdict (Legal) or Billkeeper, each (CP M86)		$ 995	$469
★ MICRO LAB, The Tax Manager		$ 250	$189
MICROPRO, WordStar+ plus free WordStar Training Manual		$ 495	$269
Mail Merge * (Call on 2 Pak, 3 Pak and others)		$ 250	$129
SpellStar *		$ 250	$129
★ WordStar Professional, 4 Pak, 3 above + StarIndex (Special)		$ 895	$429
StarIndex™	NEW!	$ 195	$109
InfoStar™		$ 495	$269
Report Star™		$ 350	$199
★ MICROSOFT, Multiplan	MSDOS	$ 275	$199
Multitool Word	MSDOS NEW!	$ 375	$275
Multitool Word with Mouse	MSDOS NEW!	$ 475	$350
Multitool, Financial	MSDOS NEW!	$ 100	$ 75
Multitool, Budget	MSDOS NEW!	$ 150	$115
muLisp muStar	NEW!	$ 250	$185
muMath muSimp	NEW!	$ 300	$225
Business BASIC Compiler	MSDOS NEW!	$ 600	$450
Pascal Compiler	MSDOS NEW!	$ 350	$260
C Compiler	MSDOS NEW!	$ 500	$375
BASIC Compiler	MSDOS NEW!	$ 395	$295
Fortran	MSDOS NEW!	$ 350	$259
COBOL	MSDOS NEW!	$ 750	$559
BASIC Interpreter	MSDOS NEW!	$ 350	$259
NORTH AMERICAN BUS SYSTEM, The Answer		$ 250	$169
OSBORNE/COMX, (Book & Disk: Business, Statistics & Math Programs)			
Some Common Basic Programs (70 programs)		$ 100	$ 69
Practical Basic Programs (40 more programs)		$ 100	$ 69
PBL CORPORATION, Personal Investor		$ 145	$ 99
PEACHTREE SOFTWARE, Peach Pak 4 (GL, AR & AP)		$ 595	$365
PERFECT SOFTWARE, Perfect Writer™		$ 389	$219
Perfect Speller™ or Perfect Calc™, each		$ 295	$129
Perfect Filer™		$ 595	$259
SELECT INFO, Select (a WPS)		$ 595	$339
SOFTWARE PUBLISHING, PFS: File	NEW!	$ 140	$ 95
PFS: Report	NEW!	$ 125	$ 85
SORCIM/ISA, SuperCalc II		$ 295	$199
SuperWriter		$ 395	$269
Spellguard		$ 195	$129
STC, The Creator		$ 300	$195
SYNAPSE, File Manager		$ 150	$ 99
SYNERGISTIC, Data Reporter		$ 250	$169
★ VISICORP, VisiCalc+ 256K		$ 250	$179
VisiDex or VisiSchedule, each		$ 250	$199
VisiTrend/Plot		$ 300	$219
VisiFile or Desktop Plan I, each		$ 300	$239

UTILITY

		LIST PRICE	OUR PRICE
★ CENTRAL POINT, Copy II PC, Backup and Utility	NEW!	$ 40	$ 35
★ COMX, Fastrak™, RAM/Disk emulator and printer spooler program. Works on any PC DOS version or RAM Card. Menu Driven.	NEW!	$ 100	$ 59
DIGITAL RESEARCH, Concurrent CP/M-86™		$ 350	$235
CBASIC 86™		$ 200	$135
CP/M 86+		$ 60	$ 40
HAYES, Smartcom II		$ 119	$ 89
MICROCOM, Microterminal		$ 100	$ 65
MICROSTUF, Crosstalk		$ 195	$129
NORTON, Norton Utilities, 14 powerful programs, 3 disks		$ 80	$ 65

HOME & EDUCATIONAL

		LIST PRICE	OUR PRICE
EPYX/Automated Simulations, Temple of Apshai		$ 40	$ 29
Oil Barons	NEW!	$ 100	$ 75
ARMONK, Executive Suite		$ 40	$ 27
BPI SYSTEMS, Personal Accounting		$ 195	$139
BRODERBUND, Apple Panic		$ 30	$ 22
CONTINENTAL, the Home Accountant Plus		$ 150	$112
★ DATAMOST, Pig Pen or Space Strike, each		$ 30	$ 22
DAVIDSON, The Speed Reader II		$ 75	$ 55
INFOCOM, Deadline		$ 50	$ 39
Zork I or Zork II or Zork III, each		$ 40	$ 29
★ INSOFT, Word Trix or Quotrix or Mystrix, each	NEW!	$ 35	$ 29
MICROSOFT, Flight Simulator (by Sub-Logic)	NEW!	$ 50	$ 35
SIERRA/ON-LINE, Ulysses & The Golden Fleece		$ 35	$ 25
PBL CORP., Personal Investor		$ 145	$ 99
SPINNAKER, Snooper Troops, #1 or #2, each		$ 45	$ 35
Story Machine or Face Maker, each		$ 35	$ 25

IBM is a trademark of IBM Corporation.

IBM-PC

System Includes
Two 320K Disk Drives by CDC
90 Day Warranty By Us
Call For Details And Configurations

HARDWARE for the IBM-PC or XT

MEMORY CHIP KITS

Add-on to your memory cards or XT mother board. 100% tested. With 90 day warranty.

		LIST PRICE	OUR PRICE
MEMORY CHIP KITS	64K Kit, 200NS, 9 chips, 4164	$ 150	$ 49
	16K Kit, 200NS, 9 chips, 4116 (for PC-1 motherboard)	$ 30	$ 16
AST	Combo Plus, 64K w/async para & clock/cal	$ 395	$ 279
	Above Combo Plus, 128K, S/P/C	$ 495	$ 375
	Above Combo Plus, 256K, S/P/C	$ 695	$ 525
★ ComX	256K RAM Card w/Fastrak™ RAM disk emulator and printer spooler software. 2 Year Warranty	$ 750	$ 225
CURTIS	PC Pedestal™ for Display on PC	$ 80	$ 65
	9 Foot Cable for IBM Keyboard (extends 3' to 9')	$ 50	$ 35
Key Tronic	KB5150, Standard Typewriter key locations NEW!	$ 269	$ 179
MICROSOFT	64K RAM Card w/Parity	$ 350	$ 250
	256K RAM Card w/Parity	$ 875	$ 625
	64K System Card, 4 Function	$ 395	$ 295
	256K System Card, 4 Function	$ 625	$ 465
MBI	Monte Carlo 64K Five Function (to 1000K)	$ 625	$ 395
Q.C.S.	Big Blue, Z80 CPU plus five functions	$ 595	$ 449
QUADRAM CORPORATION	★ Quadlink, IBM to Apple II Link	$ 680	$ 580
	Quadboard 64K, expandable to 256K, 4 function board	$ 395	$ 285
	★ Quadboard 256K, four function board	$ 595	$ 425
	Quad 512+, 64K plus serial port	$ 325	$ 275
	Quad 512+, 256K plus serial port	$ 550	$ 440
	Quad 512+, 512K plus serial port	$ 895	$ 675
	Quadcolor I	$ 295	$ 235
	Microfazer, w/Copy, Par/Par, 8K, #MP8 w/Power Supply	$ 189	$ 139
	Microfazer, w/Copy, Par/Par, 64K, #MP64 w/Power Supply	$ 319	$ 235
	Microfazer, w/Copy, Par/Par, 128K, #MP128 w/Power Supply	$ 465	$ 345
	Microfazer, Snap-on, 8K, Par/Par, Epson, #ME8 w/Power Supply	$ 179	$ 145
	Microfazer, Snap-on, 64K, Par/Par, Epson, #ME64, w/Pwr. Supply	$ 319	$ 235
	All Microfazers are expandable (w/copy to 512K) (Snap-on to 64K)		
TG PRODUCTS	Joystick	$ 65	$ 49

DRIVES

Control Data OR Tandon

		LIST PRICE	OUR PRICE
DISK DRIVES, Double Sided 360K/320K. Same as now supplied with IBM-PC & XT. Tested, burned-in. With installation instructions. 90 day warranty by us.	1 each	$ 529	$ 239
	2 or more	$ 529	$ 239
Half Height 360K or 320K Drives, same as above		$ 395	$ 199
DAVONG INTERNAL Hard Disk, (For PC-1, includes Power Supply & Controller)	5-6MB	$1995	$1195
	10-12MB	$2395	$1495
LIMITED SPECIAL	15-19MB	$2795	$1895
MAYNARD, Floppy Drive Control Board, for up to 4 drives.		$ 195	$ 165
	same with Serial Port	$ 325	$ 235
	same with Parallel Port	$ 275	$ 205

MODEMS AND ACCESSORIES

	LIST PRICE	OUR PRICE
AXLON, Datalink 1000 Hand Held Communications Terminal	$ 399	$ 299
HAYES, Micromodem II (for the Apple II)	$ 379	$ 275
Apple Terminal Program for Micromodem II	$ 100	$ 65
IBM-PC Smartcom II	$ 119	$ 89
Stock Chronograph (RS-232)	$ 249	$ 189
Smartmodem 300 (RS-232)	$ 289	$ 225
Smartmodem 1200 (RS-232)	$ 699	$ 535
Micromodem 100 (S-100 bus)	$ 399	$ 275
IBM-PC to Modem Cable	$ 39	$ 29
NOVATION, Applecat II Modem, 300 BAUD	$ 389	$ 269
212 Apple Cat, 1200 BAUD	$ 725	$ 599
SIGNALMAN, Modem MKI (RS-232)	$ 99	$ 79
SSM, Transcend 1 for Apple II Data Comm.	$ 89	$ 69
ModemCard for the Apple II	$ 299	$ 259

8" CP/M-80 BUSINESS & SYSTEM SOFTWARE - PARTIAL LIST, CALL

	LIST PRICE	OUR PRICE
ASHTON-TATE, dBase II	$ 700	$ 439
INFOCOM, Zork I or Zork II or Zork III or Starcross, each	$ 50	$ 39
MICROCRAFT, Verdict (Legal) or Billkeeper (Professional), each	$ 995	$ 469
MICROPRO, WordStar+ plus free WordStar Training Manual	$ 495	$ 269
MailMerge * (Call on 2 Pak, 3 Pak and others)	$ 250	$ 129
SpellStar *	$ 250	$ 129
WordStar Professional, 4 Pak, 3 above + StarIndex	$ 895	$ 429
MICROSOFT Multiplan	$ 275	$ 199
Fortran 80	$ 500	$ 325
BASIC Compiler	$ 395	$ 295
COBOL-80	$ 750	$ 545
BASIC-80 Interpreter	$ 350	$ 275
muLisp/muStar-80	$ 200	$ 145
M-Sort-80	$ 195	$ 145
Edit-80	$ 120	$ 80
Macro-80	$ 200	$ 145

MONITORS AND ACCESSORIES

AMDEK, 12" Green, #300G	$ 200	$ 159
12" Amber, #300A	$ 210	$ 159
★ 13" Color I, Composite	$ 379	$ 299
★ 13" Color II, RGB, Hi Res (Ap. II, III & IBM-PC)	$ 529	$ 459
★ 13" Color III, RGB, Commercial, (Ap. II, III)	$ 479	$ 399
DVM, Color II or III to Apple II Interface	$ 199	$ 175
NEC, 12" Green, Model JB1201M	$ 249	$ 159
12" Color, Composite, Model JC1212M	$ 450	$ 349
PRINCETON, RGB Hi Res	$ 795	$ 539
QUADRAM, Quadchrome 12" RGB Color	$ 795	$ 565
SANYO, 9" Green, Model DM5109	$ 200	$ 139
12" Green, Model DM8112CX	$ 260	$ 199
13" Color, Composite, Model DM6013	$ 470	$ 349
ZENITH, 12" Green, Model ZVM121	$ 150	$ 99

PRINTERS AND ACCESSORIES

EPSON MX or FX PRINTERS		CALL	CALL
LEADING EDGE, Gorilla Banana	NEW!	$ 250	$ 209
PRACTICAL PERIPHERALS, Microbuffer In-Line 64K Parallel Universal Buffer		$ 349	$ 259
Microbuffer In-Line 64K Serial Universal Buffer		$ 349	$ 259
★ STAR MICRONICS, 9x9 Dot Matrix, 100cps, 2.3K, Gemini 10"		$ 499	$ 349
9x9 Dot Matrix, 100cps, 2.3K, Gemini 15"		$ 649	$ 469
IBM-PC to Epson or Star Micronics Cable		$ 60	$ 35
Apple Interface and Cable for Epson or Gemini		$ 95	$ 59
Grappler + by Orange Micro, specify printer		$ 165	$ 119
Apple Graphics Dump Program by Epson		$ 15	$ 9
LETTER QUALITY – DAISY WHEEL PRINTERS			
COMREX, Comriter CR-1, I/F, 200 wpm	Limited Special	$1199	$ 645
Comriter Tractor Feed for CR-1		$ 118	$ 99
SMITH-CORONA, TPI Printer specify serial or parallel I/F	Special	$ 899	$ 499
TPI Tractor Feed		$ 149	$ 129

SUPPLIES: Tractor Feed Paper, Ribbons, Heads, Qume Daisy Wheels and Ribbons.

DISKETTES

CONTROL DATA CORPORATION Certified Top of the Line Diskettes.

CDC, 100 each, 5-1/4, with ring, SS, DD, 48T (Apple, IBM, etc.)	$ 550	$ 199
10 each, 5-1/4, with ring, SS, DD, 48T (Apple, IBM, etc.)	$ 55	$ 22
10 each, 5-1/4, with ring, DS, DD, 48T (IBM, H/P, etc.)	$ 75	$ 39
DYSAN, 10 each, 5-1/4, SS, SD, (Apple, etc.)	$ 69	$ 39
10 each, 5-1/4, DS, DD, 48T (IBM, H/P, etc.)	$ 89	$ 49
MAXELL, 10 each, 5-1/4, MD-1, SS, SD or SS, DD	$ 55	$ 35
GENERIK™ DISKETTES		
100 each SS, SD, 35 Track (Apple, Atari, etc.)	$ 415	$ 130
★ 1000 each SS, SD, 35 Track (Apple, Atari, etc.)	$3200	$ 995
100 each DS, DD, 40 Track (IBM, H/P, etc.)	$ 626	$ 160
★ 1000 each DS, DD, 40 Track (IBM, H/P, etc.)	$4550	$1200

With jackets, no labels, produced by a top of the line manufacturer. 90 day warranty by us.

CORVUS

6 Meg Hard Disk, w/o Interface	$2395	$1895
11 Meg Hard Disk, w/o Interface	$3195	$2695
20 Meg Hard Disk, w/o Interface	$4195	$3495
IBM-PC Interface (IBM DOS), Manual & Cable Kit	$ 300	$ 239
Mirror built in for easy backup	$ 790	$ 595
Apple Interface, Manual & Cable Kit	$ 300	$ 239
Other Interfaces, Omni-Net, Constellation, Mirror, All in Stock		

★ Means a BEST buy | AD #977

CASH & CARRY OUTLETS:
Over-the-counter sales only. Open Monday through Saturday 10:00 until 5:00.
PORTLAND, OREGON, 11507-D S.W. Pacific Hwy, Terrace Shopping Center, Tigard. On 99W between 217 and I-5. Call 245-1020.
SEATTLE, WASH., Grand Opening Soon, Call for location.

ORDERING INFORMATION AND TERMS: All Mail: P.O. Box 23068, Portland, OR 97223, Include telephone number. All items usually in stock. We immediately honor Cashiers Checks, Money Orders, Fortune 1000 Checks and Government Checks. Personal or Company Checks allow 20 days to clear. No C.O.D. Add 3% for VISA or MC. For U.S. Mainland, add 3% for shipping, insurance and handling (SI&H) by UPS with $5 minimum for SI&H. UPS ground is standard so add 3% more for UPS Blue with $10 minimum for SI&H. Add 12% total for SI&H for US Postal, APO or FPO with $15 minimum for SI&H. For Hawaii, Alaska and Canada, UPS is in some areas only, all others are Postal so call, write, or specify Postal. Foreign orders except Canada for S,I&H add 18% or $25 minimum for SI&H except for monitors add 30% or $50 minimum for SI&H. Prices subject to change and typo errors, so call to verify. All goods are new, include warranty and are guaranteed to work. Due to our low prices, ALL SALES ARE FINAL. Call before returning goods for repair or replacement. Orders received with insufficient S,I&H charges will be refunded. ORDER DESK HOURS 8 to 6 PST, Monday through Friday and 10 to 4 Saturday. 1 P.M. here is 4 P.M. in New York.

OUR REFERENCES: We have been in computers and electronics since 1958, a computer dealer since 1978 and in mail order since 1979. Banks: 1st Interstate Bank, (503) 643-4678. We belong to the Chamber of Commerce (503) 644-0123, Better Business Bureau (503) 227-0648, or call Dunn and Bradstreet if you are a subscriber. Conroy-LaPointe, Inc. is a division of O'Tech Group, Inc. Fastrak™ and Generik™ are trademarks of ComX Corporation.

THE WORLD'S LARGEST COMPUTER MAIL ORDER FIRM

CONROY-LA POINTE

CIRCLE NO. 128 ON READER SERVICE CARD Formerly **Computer Exchange**™

ALL MAIL: P.O. Box 23068, Portland, OR 97223
SHOWROOM AT 11507D SW PACIFIC HWY., PORTLAND, OR, OPEN M-SAT 10-5

VISA | master charge

NATIONAL ORDER DESK TOLL FREE
(800) 547-1289

All Other Orders Including Portland: 245-6200

Oregon TOLL FREE [800] 451-5151 | Hot Line For Information On Your Order [503] 245-1030

EPSON TECHNICAL COMPARISON

Not all EPSONs are alike; compatibility among the MX-80 III, RX-80, FX-80, and the IBM Graphics Printer

Arthur A. Gleckler

The EPSON series of printers has held a strong position in the personal computer printer market for several years, and shows no signs of decreasing in popularity or general acceptance by the computing industry. The fact that IBM sells an EPSON printer is evidence of EPSON's strength, for IBM must be convinced not only that the printer is a good product, but also that the company that produces it is financially sound.

EPSON printers are versatile, inexpensive, and rugged, and have set the standard for print quality, flexibility, and compatibility. This last characteristic is very important to the software developer, for without compatibility, designing programs to be portable from printer to printer would be a formidable task.

Even within the EPSON series, however, there are inconsistencies. While one printer will allow user-defined characters and will print proportionally, another has only limited graphics ability and no advanced printing features. The reason for these variations lies with the printer control codes, which govern which print style is in use, the direction of printing, the size of print, etc. In order to accommodate the differences among the printers, it is necessary to find control codes that will perform the same function on all of the printers.

The fact that IBM sells an EPSON printer is evidence of EPSON's strength, for IBM must be convinced not only that the printer is a good product, but also that the company that produces it is financially sound.

This article discusses the capabilities of four EPSON printers: the MX-80 version III (or MX-80 with Graftrax Plus), the new RX-80 and FX-80 printers, and the IBM Graphics Printer. The article also explains the differences in printer control codes for each machine, and concludes with some suggestions for making programs compatible with all of the EPSON printers.

Overview of the Four Printers
The MX-80 printer, the oldest and most numerous printer in the EPSON line, is no longer being sold directly to consumers. It is now produced only to fulfill contracts with OEMs (Original Equipment Manufacturers) who sell the MX-80 under their own labels.

The IBM Graphics Printer is a prime example of this. Although it is slightly different from the MX-80, the IBM printer is essentially an MX-80. It is doubtful that EPSON will continue supplying MX-80s to OEMs after the current contracts expire; the RX-80 and FX-80 printers will probably replace it, and IBM may begin to sell the new printers sometime in the not-too-distant future.

The RX-80 printer is an addition to the EPSON line. It is very similar to the MX-80, but 25 percent faster (100 characters per second as compared to 80 cps on the MX-80). It also has many of the advanced features of the faster FX-80.

The FX-80 is another new printer from EPSON. Running at 160 cps, it is twice as fast as the MX-80 and has a larger print buffer, more powerful graphics features, user-definable character sets, and international character sets. The FX-80 is a statement of EPSON policy: it has new features, more advanced than its predecessors, but still preserves the standard features of the EPSON line.

Printer Control Codes

The print mode controls are largely compatible in the EPSON lines. The condensed, double-strike, emphasized, enlarged (double-width), subscript, superscript, and underlined modes operate identically when invoked individually. However, the FX-80's ability to set multiple print modes with one ESCape sequence is a new feature not present on the other printers. Most print modes can be accessed in a standard format, with the exception of the alternate, elite, and proportional modes.

The EPSON printers allow using several print modes at one time. For example, enlarged, double-strike, and emphasized modes can be combined to produce large, dark characters. There are three combinations of print modes that are not allowed on any of the EPSON Printers: elite condensed, elite emphasized, and condensed emphasized; any other combination is legal.

All four printers have 480- and 960-dot bit image graphics capability. However, the RX-80, FX-80, and IBM Graphics printers have other graphics capabilities. For example, all three can operate at many different speeds in the graphics modes.

One problem that has plagued users of EPSON graphics is the unusual spacing of the dots. When a program sends the printer codes that should produce a square (i.e., equal number of dots are used for all sides), the earlier printers produce a rectangle instead. This is because of the difference between the vertical and horizontal spacing of the dots. The FX-80 corrects this problem by providing a new graphics mode in which the dots are evenly spaced.

There are two standard paper feed functions in the EPSON line: line feed and perforation-skipping. The line feed advances the paper one line, while the perforation-skip command tells the printer to advance past the perforations every *n* lines. The printer cannot sense the presence of a perforation, but because it knows how many lines there are on one page, it can count until it reaches a perforation and then skip it.

The most interesting nonstandard paper feed function is, again, present only on the FX-80. The reverse feed allows the printer to move paper that has already been printed backwards under the print head. Note that a reverse paper feed will only work in certain areas of the paper; if the print head has just passed a perforation, the paper may get caught in the printer.

All of the EPSON printers allow the setting and using of horizontal and vertical tabs, form feeds, and form length in lines and inches. One peculiar change made to the IBM Graphics printer is that it will not recognize the backspace (code 8) character. In all of the other EPSON printers, the backspace causes the contents of the print buffer to be printed and the print head to be moved over the character that was last in the print buffer. For example, if you type this in BASIC command mode:

```
LPRINT "abcde";CHR$(8);
"fghij"
```

using an MX-80, RX-80, or FX-80, the result will be

abcdefghij

with the 'f' and 'e' characters overlapping. However, on the IBM Graphics printer, the same sequence would result simply in:

abcdefghij

as the printer ignores the backspace.

Nonstandard format control functions include column width set, left margin set, and vertical format unit set. The vertical format unit is a commercial feature unique to the FX-80; it allows setting of vertical tabs in independent fields on the page. In other words, two sets of vertical tabs can be used independently or at the same time on one sheet of paper.

Input data control on the printers allows the automatic setting of the high bit (bit 7) on input, clearing the print buffer, and printer select and deselect. There are no standard functions in this category.

Another entirely nonstandard category of functions is the download character set control; only the FX-80 has these features. The download character set ability allows the user to define or redefine the characters that will correspond to specific codes. This feature can be simulated on all of the other printers (and the FX-80) through the use of bit-image graphics (although this method is limited by the need for more programming and its slow speed).

All four printers support the bell (character code 7) and carriage return. Other standard features include selection of the paper-end sensor and selection of unidirectional printing for one line or until deselected.

How to Stay Compatible

Software developers have three choices in dealing with the problem of preserving printer compatibility:

1) They can avoid all but the most trivial capabilities of each printer, using only displayable ASCII characters and the default print modes.
2) They can choose to conform to the characteristics of one printer only, hoping that their programs will work on others' printers.
3) They can allow the user to change the codes with which the program accesses the printer.

The first option is the most widely-used method, mainly because most programs do not need the more advanced capabilities. While a terminal program need not

know how to work with bold and underlined text on the printer, a word processor may; this method is not suitable for all programs.

The second method is also popular. Fortunately for the owners of the EPSON printers, many software houses have blessed the MX-80, and almost all commercial programs support the standard EPSON capabilities. However, the advanced features of the EPSON line are far from standard and, not everyone owns an EPSON.

The third option, while its flexibility makes it clearly the most desirable for the end user, can be a tedious and painstaking programming job. Accounting for all the different printer code formats and printer capabilities while maintaining simplicity for the user is a difficult job.

Rules of Thumb

There are several rules of thumb that should be followed to preserve compatibility.

First, a software developer should never use graphics modes other than standard 480- and 960-dot modes. All four printers use the same format for these modes, and the newer graphics modes offer few advantages that cannot be simulated on the less advanced printers.

It is also not wise to use download character sets without providing other methods of producing the same characters. For example, if an APL program is designed to output all of the APL language characters to an FX-80 printer using the download character set capability, it should also be able to use the graphics modes to produce the same characters on the MX-80 (which does not have download character set ability). Thus, FX-80 users will receive the benefits of speed that a download character set can provide, while MX-80 users will not be punished for buying their printers earlier.

Because horizontal and vertical tabs, margins, and column width can be controlled entirely through software, there is truly no reason to risk losing compatibility by using these functions.

The print mode function category is almost entirely standard. Although there are nonstandard ways of achieving standard modes (i.e. using ESC SO to set enlarged mode, when SO works fine), most functions can be accessed on all of the printers by a single control code. There are exceptions, however.

With the FX-80, EPSON has introduced two new print modes: elite and proportional. In proportional mode, characters are spaced according to actual width rather than for exactly 80 columns per line. This produces the illusion of typesetting and, according to some, makes the print easier to read. Because of the peculiar spacing in proportional mode, the software to right-justify a line of text becomes much more complex than in normal print modes. Proportional print is not useful for many tasks other than word processing. Although word processing software *will* have to deal with the many problems of compatibility in this field, few other programs will need proportional print. Rule of thumb: avoid it, if possible.

While the IBM Graphics printer's ability to produce the complete IBM character set is useful, it is also unique. Unfortunately, programs written to take advantage of this ability will work *only* on the IBM Graphics printer. Therefore, software developers should avoid using this feature and use other methods to print the IBM character set.

There are many problems to be considered in making programs function well with the world of incompatible printers. However, with careful consideration and good programming, most of these obstacles can be avoided. ■

Printer Technical Comparison Chart

Abbreviation key to tables 1 and 2

ESC	=	Escape (27 or 155)
ba	=	Integer in {0,1} (boolean value)
bl	=	Integer in {0, 1, 48, or 49} (boolean value 0 or 1)
by	=	Integer between 0 and 255 inclusive
fn	=	Integer between 0 and 85 inclusive
gm	=	Integer between 0 and 6 inclusive
hb	=	High byte
hn	=	Integer between 1 and 16 inclusive
lb	=	Low byte
ln	=	Integer between 1 and 127 inclusive
og	=	Integer in {0, 1, 2, 3, 4, 6}
sn	=	Integer between 0 and 63 inclusive
tt	=	Integer between 1 and 22 inclusive
vn	=	Integer between 0 and 7 inclusive
ze	=	Integer between 0 and 8 inclusive
zt	=	Integer between 0 and 10 inclusive
..	=	Repeated

Table 1

Functions common to EPSON MX-80 III, EPSON RX-80, EPSON FX-80, and IBM GRAPHICS printers:

Function	ASCII Code	Decimal Code
BIT IMAGE GRAPHICS		
Set 480-pin bit-image mode	ESC K lb hb by..	27 75 lb hb by..
Set 960-pin bit-image mode	ESC L lb hb by..	27 76 lb hb by..
FORMAT CONTROL		
Horizontal tab	HT	9
	HT	137
Vertical tab	VT	11
	VT	139
Form feed	FF	12
	FF	140
Set form length in lines	ESC C ln	27 67 ln
Set form length in inches	ESC C NUL tt	27 67 0 tt

LINE SPACING		
Set line spacing to 1/8-inch	ESC 0	27 48
Set line spacing to 7/72-inch	ESC 1	27 49
Set line spacing to 1/6-inch	ESC 2	27 50
Set line spacing to by/216-inch	ESC 3 by	27 51 by
Set line spacing to fn/72-inch	ESC A fn	27 65 fn
MISCELLANEOUS		
Bell	BEL BEL	7 135
Carriage return	CR CR	13 141
Paper-end detect deselect	ESC 8	27 56
Paper-end detect select	ESC 9	27 57
Print left-to-right for one line	ESC <	27 60
Unidirectional print start/end	ESC U bl	27 85 bl
PAPER FEED EXECUTION		
Line feed	LF LF	10 138
Perforation-skip mode set	ESC N ln	27 78 ln
Perforation-skip mode cancel	ESC O	27 29

PRINT MODES

NOTE: All print mode combinations are legal *except the following*:
Elite Condensed
Elite Emphasized
Condensed Emphasized

Condensed mode cancel	DC2 DC2	18 146
Condensed mode set	SI SI	15 143
Double-strike mode cancel	ESC H	27 72
Double-strike mode set	ESC G	27 71
Emphasized mode cancel	ESC F	27 70
Emphasized mode set	ESC E	27 69
Enlarged mode cancel	DC2 DC2	20 148
Enlarged mode set/cancel	ESC W bl	27 87 bl
Enlarged mode set	SO SO	14 142
Superscript/subscript mode cancel	ESC T	27 84
Superscript/subscript mode set	ESC S bl	27 83 bl
Underlined mode set/cancel	ESC - bl	27 45 bl

Fortunately for the owners of the EPSON printers, many software houses have blessed the MX-80, and almost all commercial programs support the standard EPSOM capabilities.

TABLE 2

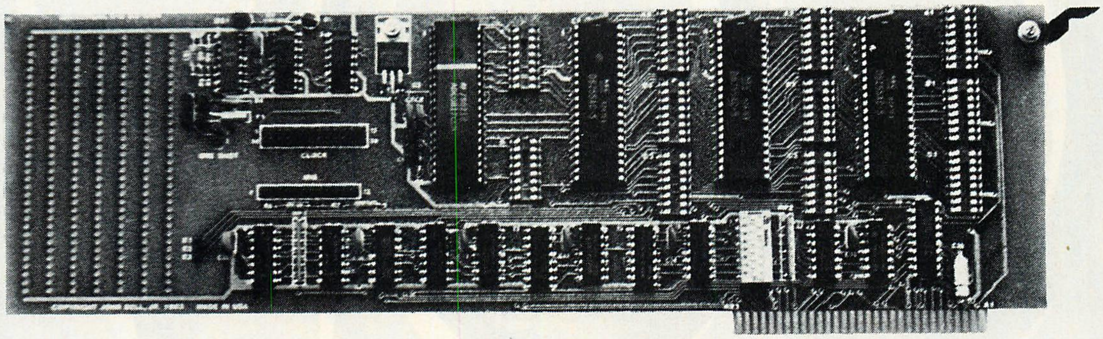
Functions differing among EPSON MX-80 III, EPSON RX-80, EPSON FX-80, and IBM GRAPHICS printers:

NOTE: Decimal values are given below ASCII codes.

Function	EPSON MX-80 III	EPSON RX-80	EPSON FX-80	IBM GRAPHICS
BIT IMAGE GRAPHICS				
8-pin mode set		ESC * og hb lb by.. 27 42 og hb lb by..	ESC * gm hb lb by.. 27 42 gm hb lb by..	
9-pin mode set			ESC ^ bl hb lb by.. 27 94 bl hb lb by..	
Double speed, 960-dot mode		ESC Y hb lb by.. 27 89 hb lb by..	ESC Y hb lb by.. 27 89 hb lb by..	ESC Y hb lb by.. 27 89 hb lb by..
1920-dot mode		ESC Z hb lb by.. 27 90 hb lb by..	ESC Z hb lb by.. 27 90 hb lb by..	
DOWNLOAD CHARACTERS				
Copy ROM fonts to download area			ESC : NUL NUL NUL 27 58 0 0 0	
Define download character generator			ESC & NUL by.. 27 38 0 by..	
Select internal/download character generator			ESC % ba NUL 27 37 ba NUL	

FORMAT CONTROL				
Backspace	BS 8	BS 8	BS 8	
Column width set	ESC Q by 27 81 by	ESC Q by 27 81 by	ESC Q by 27 81 by	
Horizontal tab set	ESC D by..NUL 27 68 by..0	ESC D by..NUL 27 68 by..0		ESC by..NUL 27 68 by..0
Left margin set		ESC l by 27 108 by	ESC 1 by 27 108 by	
Vertical tab set			ESC B by..NUL 27 66 by..0	ESC B by..NUL 27 66 by..0
VFU channel select			ESC / vn 27 47 vn	
VFU position set			ESC b vn by..NUL 27 98 vn by..0	
INPUT DATA CONTROL				
Alternate character set selection			ESC 6 27 54	ESC 6 27 54
Bit 7 = one	ESC > 27 62		ESC > 27 62	
Bit 7 = zero	ESC = 27 61		ESC = 27 61	
Bit 7 control cancel	ESC # 27 35		ESC # 27 35	
Clear buffer			CAN 24	CAN 24
Clear last byte in buffer	DEL 127 DEL 255	DEL 127	DEL 127 DEL 255	
Control code selection			ESC 7 27 55	ESC 7 27 55
Printer deselect			DC3 19	
Printer select			DC1 17	
Select undefined area as control/ printable			ESC I 27 73	
MISCELLANEOUS				
Half-speed printing cancel/select		ESC s bl 27 115 bl	ESC s bl 27 115 bl	
Incremental mode set			ESC i bl 27 105 bl	
Initialize printer	ESC @ 27 64	ESC @ 27 64	ESC @ 27 64	
International character set select		ESC R zt 27 82 zt	ESC R ze 27 82 ze	
PAPER FEED EXECUTION				
Paper feed		ESC J by 27 74 by	ESC J by 27 74 by	ESC J by 27 74 by
Reverse feed			ESC j by 27 106 by	
PRINT MODES				
Alternate mode deselect	ESC 5 27 53	ESC 5 27 53	ESC 5 27 53	
Alternate mode select	ESC 4 27 52	ESC 4 27 52	ESC 4 27 52	
Condensed mode set		ESC SI 27 15	ESI SI 27 15	
Elite-sized print mode select		ESC M 27 77	ESC M 27 77	
Enlarged mode set		ESC SO 27 14	ESC SO 27 14	
Normal print mode select		ESC P 27 80	ESC P 27 80	
Print mode select			ESC ! sn 27 33 sn	
Proportional spacing set/ cancel			ESC p bl 27 112 bl	

CONTROL

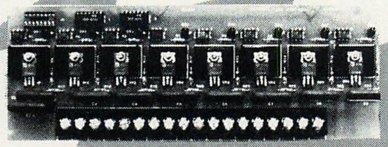


FEATURES:

- 16 8 BIT ANALOG INPUTS (0 TO 5 VOLTS)
- 9 8 BIT PARALLEL PORTS (72 LINES)
- PROTOTYPING AREA
- TIMER CLOCK 32768 HZ. TO 512 SEC.
- INTERRUPT CIRCUITRY
- LED FOR POWER AND CLOCK
- GOLD EDGE CONNECTOR
- SOLDER MASK BOTH SIDES
- LOW POWER LESS THAN 1.5 WATTS
- 16 PIN RIBBON CONNECTORS

This board is designed to be an economical interface for data acquisition and control applications. The 16 Analog inputs can measure voltage from zero to five volts or temperature using sensors. The nine Parallel ports can be set for input or output. All the ports connect through 16 pin ribbon cables. This board is compatible with other JOHN BELL ENGINEERING INC. products.

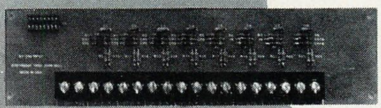
ORDER PART NUMBER 83-064A $299.95



120 VAC OUTPUT CONTROL

This board has eight optically isolated triac switches. Each switch can control 200 watts. It connects via a 16 pin ribbon cable to a parallel output port. Screw terminals are provided for 120 VAC connection.

ORDER PART NUMBER 82-332 $119.95



INPUT PROTECTOR

This board protects the inputs of the ANALOG input or PARALLEL input port. There are 4.7k pullups, diodes and caps for each line. It connects via a 16 pin ribbon cable. Screw terminals are provided for external connection.

ORDER PART NUMBER 82-334 $89.95

JOHN BELL ENGINEERING, INC.

ALL PRODUCTS ARE AVAILABLE FROM **JOHN BELL ENGINEERING, INC.** • 1014 CENTER ST., SAN CARLOS, CA 94070
ADD SALES TAX IN CALIFORNIA • ADD 5% SHIPPING & HANDLING 3% FOR ORDERS OVER $100
10% OUTSIDE U.S.A.
ADD $1.50 FOR C.O.D.

(415) 592-8411
WILL CALL HOURS: 9am - 4pm

MasterCard VISA

Load and Go: .COM and .EXE Files Explained

Tom Carter

PC DOS supports two distinct types of loadable and executable program modules: COM and EXE, each with advantages and disadvantages.

The Sample Program

DPFILE.ASM is a utility that sequentially reads and displays (dumps), in a neatly formatted output, the contents of each physical sector of a specified source file. DPFILE can be used to dump any PC DOS file and is invoked like an standard DOS command:

A)DPFILE
File name = B:COMMAND.COM

LIST-1 is the assembly language listing for the EXE version and LIST-2 is the listing for the COM version. SETUP-1 and SETUP-2 illustrate the program prep-

Tom Carter develops utility, networking, and telecommunications software for the IBM PC and other microcomputers. His company, Real Time Concepts, is in Morgantown, IN.

aration steps required for each format.

Two output listings from DPFILE illustrate the actual content and structure of EXE and COM program files. DUMP-1 is the output of program DPFILE.EXE and DUMP-2 is the output of DPFILE-.COM. (SET-UPS, DUMPS, and LISTS follow this article.)

The dump for each block is preceded by the display of the relative block number and terminated with a Form Feed character.

This form feed makes each block dump occupy a single printed page in cases where CTRL/PRTSC is being used.

The dump displays a hexadecimal and decimal byte offset counter on the left, the hexadecimal representation of the file contents in the center (16 bytes per line), and the ASCII representation on the right. All unprintable characters are represented by ASCII periods.

Linking and Loading

An assembler or compiler transforms a source program file into an object file (OBJ extension). An object file contains information for the LINK program as well as the machine language representation of the source program's instructions and statements. From this information and machine language data, LINK creates a loadable and executable EXE file as its primary output.

A portion of COMMAND-.COM, called the loader, is responsible for performing the functions of program loading. When an executable file name is entered, the loader reads the program file from mass storage into memory. It then prepares the program for execution by initializing specific hardware registers and relocating certain address references. Finally, the loader passes control to the program's start address and the program executes.

One of the primary tasks in the loading procedure is the relocation process. In LIST-1, the 4 digit numbers on the left are the relative addresses allocated to instructions and data as defined by the source statements. These addresses are relative to the beginning of the program and have no relation to where the program physically loads.

Part of the loading process is locating an unoccupied area of memory large enough to hold the program. The first address of this area is designated as the base load address. When the program loads, the addresses from the listing are calculated relative to this base address. Each segment contained in the loaded program is identified by its own base address. References to these base addresses must be recalculated relative to the load address. To do this correctly, the loader must know where the references are, as well as the segment offset values.

LINK is responsible for calculating and storing this relocation information as a part of an EXE program file. Each relocatable reference (item) is composed of 4 bytes—2 bytes pointing to the reference and 2 bytes containing the segment offset to be relocated.

The EXE File

An EXE file contains a header, which begins in the initial block. This header is at least 512 bytes long but can be longer if a sufficient number of relocatable items exist; it contains the information used by the program loader. Included in this information are the following data (refer to DUMP-1):

1)bytes X'0' and X'1' contain an ASCII "MZ". These two characters are the LINK program trademark and signify that the file has been linked and is in the proper EXE format.

2)bytes X'2' and X'3' contain a binary count of the bytes used in the last block of the EXE file. In our example, these bytes contain X'C800'. The least significant bits in this count are represented by the left byte and the most significant bits are in the right byte. Therefore, in our example, there are only 200 bytes (X'C8') used in the last 512 byte block. Refer to DUMP-1 to see the last block. This byte count is used by the loader to calculate the size of the EXE program.

3)bytes X'4' and X'5 contain a binary count of blocks in the EXE file. This count includes the header and is also used by the loader to calculate the actual size of the EXE program. In our example, these bytes contain X'0400' indicating that the EXE file is 2048 bytes (4*512) long.

4)bytes X'6' and X'7' contain a binary count of the relocation items located in the EXE file header. In our example, there is only one.

5)bytes X'8' and X'9' contain a binary count of the paragraphs (a paragraph is a term describing an entity 16 bytes in length and aligned on a 16 byte boundary) required to hold the EXE file header. Our sample contains X'2000' in these bytes. This sizes the header as 20 hexa-decimal paragraphs or 512 bytes (32*16) in length.

6)bytes X'A' and X'B' and X'C' and X'D' contain counts of the minimum and maximum number of paragraphs which must reside above (higher in memory) the end of the loaded program. These bytes are used jointly to properly position a program in memory. Bytes X'A' and X'B' contain the minimum count of paragraphs. In our example, no paragraphs are required above the loaded program.

> *One of the primary tasks in the loading process is the relocation process. LINK is responsible for calculating and storing this information.*

7)bytes X'C' and X'D' contain the maximum count of paragraphs required. A value of X'FFFF' in these bytes is a special indicator

signifying that as many paragraphs as possible are to exist above the loaded program. This forces the program to load as low as possible in memory. If bytes X'A' and X'B' and X'C' and X'D' contain zeros, the program is linked to load as high as possible in memory.

These bytes are called the HIGH/LOW load switch and are affected by the /H LINK parameter. Our sample program is linked for low-memory loading. High memory

Bytes X'12' and X'13' are used by the COMMAND.COM loader as a checksum value for the EXE file. This checksum value is the negative sum of all words in the EXE file.

loading is typically used when creating routines which are called from BASIC programs. Loading a callable routine high usually positions it outside of any memory used by the BASIC interpreter or by any BASIC program. Refer to the LINK section in the DOS manual and to the BASIC language manual for more information.

8)bytes X'E' and X'F' contain the segment offset of the stack segment. In our example, these bytes contain X'1900'.

In the EXE version of the source program (List 1), the program stack is defined by the following statements:

STACKS SEGMENT PARA
STACK 'STACK'
DB 50 DUP('$$')
STACKS ENDS

A segment must load on a paragraph boundary (modulo 16), and the segment address (also modulo 16), is expressed without the final digit, which is always zero. Therefore, in our example, X'19' is the relative segment offset. Byte offset X'0190' in of DUMP-1 shows this stack segment allocation. The loader adds this value to the actual load segment address (called the start segment) and sets the result into the SS segment register before passing control to the EXE program.

9)bytes 'X10' and X'11' contain the value to be placed into the SP (stack pointer) register before the loader passes control to the EXE program. In our example, the stack is 100 ('64') bytes in length. The X'64' located in these bytes points the SP register to the end of the sample stack segment. Data is "pushed down" onto the stack and "popped up" from it. A push decrements the SP register, combines it with the SS register to form an absolute address, and then pushes one word (2 bytes) of data into the location designated by the address. A pop combines the SP and SS registers, pops the data from the resulting location, and then incrememts the SP register.

10)bytes X'12' and X'13' are used by the COMMAND.COM loader as a checksum value for the EXE file. This checksum value is the negative sum of all words in the EXE file. After the program is loaded into memory, the checksum value is recalculated. If the checksum values differ, the program is not executed.

11)bytes X'14' and X'15' contain the value to be placed into the IP (instruction pointer) register when the EXE program is given control. In our sample, this is at byte offset zero. DUMP-1 contains the actual machine code for the EXE program beginning at this offset.

12)bytes X'16' and X'17' contain the offset of the code segment of the EXE program. Our example has a zero for this value. Before control passes to the loaded program, this segment offset is added to the start segment address (where the program begins) and the resulting value is moved to the CS register. Combining the CS segment register and the IP register results in the absolute starting address of the program.

13)bytes X'18' and X'19' contain the byte offset of the first relocation item in the header block. Our sample offset is X'1C00', pointing to the X'OA' at byte offset X'001C' in the dump.

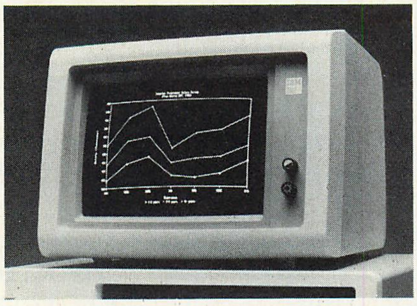
A relocation item consists of two 2 byte fields. The first field points (as a byte offset) to a word requiring relocation in the load module. The second field contains a segment value which must be added to the current contents of the specified word to accomplish the relocation task.

The instruction that contains the word to be modified is in of DUMP-1. Bytes X'A' and X'B' contain the relative segment address which must be relocated in relation to the start segment address. The assembly language instruction which corresponds to these locations can be found in List-1 at relative offset X'9'. The data segment, in DUMP-1, starts at offset zero. If the middle blocks of the dump were combined, the data segment would be at offset X'0200'. This is a segment offset of X'20' (the value contained in locations X'A' and X'B') and is the value to be relocated.

14)bytes X'1A' and X'1B' contain the overlay number. A zero in these locations indicate that the header refers to the resident portion of the EXE program.

15)the remaining 484 bytes in this block are dedicated to the two 2-byte relocation items. Thus, 121 of these items may be accommodated in the first header block. If there are more than 121, additional header blocks will be required.

In the original assembly language listing, the stack segment is defined first, (STACKS), the data segment second (WSTORE) and the code segment last (CODSEG). In the dump, these segments are rearranged: the code segment is first,



Getting 1·2·3™ to do graphics on an IBM® Monochrome takes a lot…

Exactly one Hercules Graphics Card.™

What a lot of people have learned the hard way is that you can't run 1-2-3's graphics on the IBM PC monochrome display. That is, until now.

Now, with the Hercules Graphics Card, you *can* run 1-2-3's graphics on the monochrome display. (Of course, 1-2-3's spreadsheet and information management functions work too.)

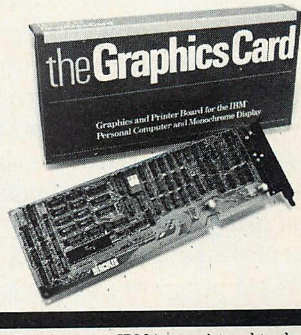
But naturally the Hercules Graphics Card does a lot more than just make a great program like 1-2-3 better. It gives you high resolution bit mapped graphics on the monochrome display—something the IBM monochrome card simply can't do.

The Hercules Graphics Card replaces IBM's monochrome card giving you text compatibility *or* two pages of graphics, each with a resolution of 720h × 348v. You'll get the monochrome's high quality character set, software compatibility with programs like VisiCalc,® WordStar® and dBase II,™ and a printer interface for any parallel port printer.

Included with each Hercules Graphics Card is software that enables you to use the PC's BASIC graphics commands. Or you can program in other languages and use Graph X,™ our set of assembly language graphics subroutines. (Graph X is available now for only $50 and includes a graphics screen dump.)

The Hercules Graphics Card. Only **$499.** A small price to pay for graphics on the monochrome display. Call or write for the name of the dealer nearest you.

Hercules Computer Technology
2550 Ninth Street
Suite 210
Berkeley, CA 94710
(415) 540-6000



HERCULES

Graphics software requiring the IBM Color Graphics Adapter is not compatible with the Hercules Graphics Card. Hercules compatible software from other vendors includes programs for business graphics, CAD, and word processing with scientific characters. Call or write for more information.

Hercules Graphics Card and Graph X are trademarks of Hercules Computer Technology. 1-2-3 is a trademark of Lotus Development. IBM is a registered trademark of IBM. VisiCalc is a registered trademark of VisiCorp. WordStar is a registered trademark of MicroPro. dBase II is a trademark of Ashton-Tate.

CIRCLE NO. 151 ON READER SERVICE CARD

followed by the stack and data segments. The code segment ends at offset X'0189' and the next 7 bytes contain nulls. Since a segment must begin on a paragraph boundary, the stack cannot start until the next modulo 16 address, which is X'0190'. The stack is defined as 100 bytes in length and the stack segment ends at offset X'01F3'. The next 12 bytes are wasted; a stack should always be allocated in multiples of 16 bytes to avoid this waste.

Com Files

A COM file is created by the EXE2BIN utility program. EXE2BIN uses an EXE program file as its input and creates, as its output, a binary image of the executable program with a BIN extension. For program execution, the BIN file must be renamed with a COM extension.

Relocatable references are not allowed in an EXE program which is to be converted to a COM program. Therefore a COM file does not require a file header to define relocation items or the initial contents of the CS, SS, or SP registers. The maximum size for a COM file, however, is limited to one segment, or 64K in size. The loader initializes each segment register to the address of the start segment. EXE programs, on the other hand, are initialized with the ES and DS segment registers pointing to the program segment prefix, the CS register pointing to the code segment, and the SS register pointing to the stack segment.

A COM program is loaded by first locating a starting segment. The program segment prefix, which is X'0100' bytes long, is then built beginning at this address. The COM program, based on the size contained in its directory entry, is read into memory beginning at the end of the program segment prefix. A COM program must be coded so that its start address is at byte offset X'0100'. This is accomplished by using an ORG assembler pseudo-op. Refer to the beginning of LIST-2 for an example.

Once a COM program is loaded into memory, passing control to it requires that all segment registers be initialized to the address of the program segment prefix (start segment), X'0100' be moved to the IP register, and the SP be initialized to either the bottom of the COMMAND.COM transient area or to the end of the COM program's segment, depending on which is lower in memory.

Since COM executable files load and initialize faster and are more compact than EXE files, it seems reasonable that all assembled or compiled programs would be made into COM programs. However, the 64K byte program size limitation, as well as the relocatable reference prohibition, excludes many programs. When these constraints don't exist, an effort should be made to utilize the benefits of COM formatted files. To do this, you must know the implementation differences between EXE and COM files. These differences occur at two levels - the coding level and the preparation level.

Coding Level Differences

At the coding level, the utilization of multiple segments must be carefully considered since relocatable references are not allowed. A provision for stack handling must also be formulated since a stack segment is not allowed. In coding a program, I usually allocate a separate segment to each portion of the program - the code, the stack, and the data. LIST-1 illustrates the format and sequence I use.

To convert DPFILE to its COM compatible format requires the following coding changes:

1)First, the relocatable reference in LIST-1 must be eliminated. This reference

MOV AX,WSTORE

moves the data segment address into the DS register as the program initializes. The most efficient way to eliminate the reference is to eliminate the data segment and incorporate it into the code segment. A COM program initializes with all segment registers pointing to the same address—the program segment prefix. All addressing references must be relative to this address. This makes the coding task easier since segment register manipulation need never occur.

This is particularly helpful if the program segment prefix structures (2 file control blocks and a 128 byte disk transfer area) are utilized. Look at FILE SPEC PASSED in LIST-1. Four of the first nine instructions on this page are eliminated. In addition, the extra segment designation (ES:) on four of the instructions at the bottom of the page are no longer necessary.

2)The next task is to eliminate the stack segment and formulate the stack handling methodology. The actual elimination is easy—simply delete the lines from the source file. Remember, in the EXE version, the loader initializes the SS segment register to the stack segment address and the SP register to the end of the stack.

A COM file is created by the EXE2BIN utility program. EXE2BIN uses an EXE program file as its input and creates, as its output, a binary image of the executable program with a BIN extension.

In the COM version, the SS register contains the address of the program segment prefix and the SP register is set to the end of the program's segment or to the end of the transient portion of COMMAND.COM. The size of the stack is an unknown quantity.

To control this, the SS segment register is left pointing to the program segment prefix. A value of X'0100' is moved to the SP register during initialization. This positions the end of the stack to the end of the program segment prefix. The first file control block is used by the program, so the length of this stack is the accumulated sizes of the second file control block and the disk transfer area—a total of 83 words. Almost any program is accommodated by a stack this large.

3)Another constraint confronting COM programs is the requirement to have X'0100' as the first location and starting instruction of the program. At this point in our conversion, the stack and data segments are eliminated and only a code segment (containing code and data) remains. The sequence of the EXE version coding coupled with the inability of the editor (EDLIN) to move source statements make it a time-consuming task to totally reorganize the source file.

An EXE file begins with a header, which contains the data required by the loader to relocate certain address references at execution time.

A simpler and faster solution, as LIST-2 shows, follows these steps:

a)declare the code segment immediately following the program TITLE statement.

b)use an assembler ORG macro pseudo-op to position the location counter to X'0100'.

c)define the program starting address with a label and PROC statement.

DPFILE PROC FAR

d)declare the ASSUME statements pointing each segment register to the program segment prefix.

e)code the initial instruction as a branch (JMP) to the actual body of the coding section (bypassing the data statements).

f)leave the data allocation statements in place.

g)define a new label (SSTART) preceding the coding section. This label is used as the destination of the branch instruction in step e.

h)declare the END statement as the last statement in the source program. The label of the starting address (step c) should also be defined as part of the END statement.

END DPFILE

Preparation Level Differences

The first step in creating both types of files is to assemble the source program and create an object file. The object file is then linked, creating an EXE file. Linking the EXE version creates the executable EXE file. Linking the COM version is only an intermediate step and causes the error message

Warning: No stack segment

to appear. This is only a warning; recall that the rule is a COM file cannot have a stack segment. The EXE file of the COM version is not the end result anyway as is not meant to be executed.

Once the COM version is linked, two additional steps are required to produce an executable COM file. First, the EXE file of the COM version is processed by EXE2BIN to produce a binary image of the executable program. In essence, this process discards the first X'0100' locations of the EXE file and writes the remaining data as a BIN file. When the program loads, these first X'0100' locations are used in constructing the program segment prefix. Since all segment registers initially point to the prefix, offset X'0100' points to the program's start address. If EXE2BIN detects a stack segment or any relocatable reference, the process aborts and an appropriate error message appears. The second additional step is renaming the BIN file to a COM file. After completing this, the COM file is ready for execution. SETUP-1 and SETUP-2 clearly illustrate the preparation procedures of both file versions.

Pros and Cons

A COM program file is physically smaller than an equivalent EXE file since an information header isn't required. COM files have the advantage of requiring less disk storage and, probably, less memory. Since relocation doesn't occur and there is no file header to be read and manipulated, COM files also have the advantage of loading faster.

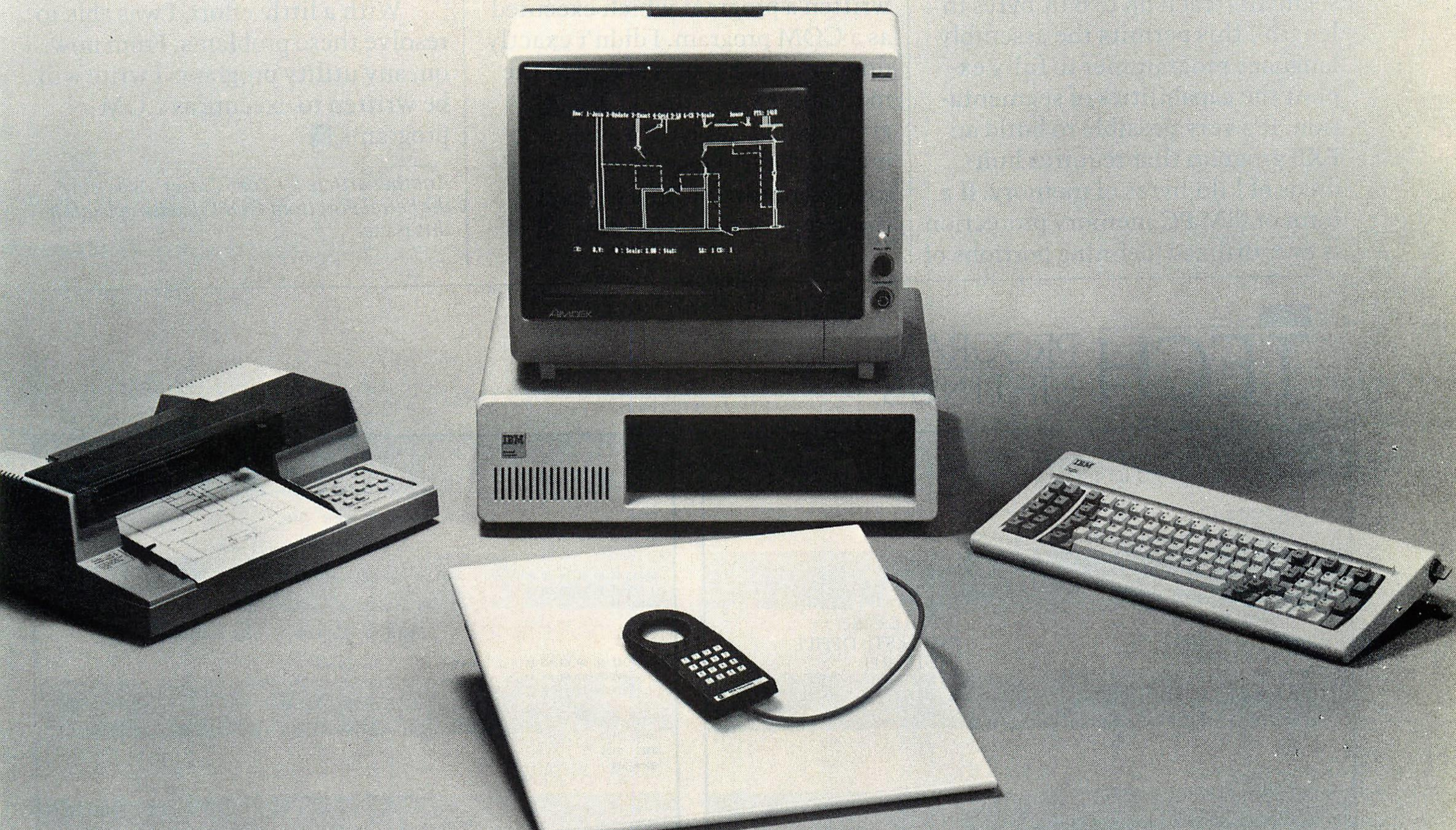
Another advantage of COM files is their ability to load in any available location in memory. This is particularly pertinent if multiple BASIC callable routines must reside simultaneously in memory. Linking the equivalent EXE routines high allows only one in memory at a time since they all load at the same high address. This potentially requires BLOADing each module when it's needed. COM files, because of their ability to load anywhere, could be selectively positioned in memory allowing multiple COM programs to reside simultaneously in memory.

COM programs have the disadvantages of requiring stricter (albeit simpler) coding methods and of limiting the utilization of segmentation, since COM files may not contain relocatable items. No stack segment may be specified and the COM program (including instructions and data) cannot exceed 64K bytes.

EXE program coding is more flexible and less confining than COM program coding program and more closely resembles the stan-

Add CAD to your PC

For less than you might think!



VECTOR SKETCH™ by GTCO Corporation

Vector Sketch is a 2D graphics package that runs on your IBM PC. Included is an electromagnetic digitizer (sizes range from 11" × 11" to 42" × 60") for fast, efficient, data entry. Drawings are stored on disk, and can be recorded on a matrix printer or drawn by a pen plotter.

Business Applications
- Space Planning
- Flow Charts
- Graphic Design
- Illustrations
- Forms Design
- Product Design

Technical Applications
- Architectural Design
- Mechanical Design
- Electrical Design
- Drafting
- Mapping
- Seismic Digitizing

Functions (Partial List)
- Rubberbanding
- Symbol file
- Figure generation
- Labeling
- Zoom, Pan & Scale
- Exact, Grid & Join
- Select line style
- Plot, Print & Store
- Erase line or symbol
- Display X-Y
- Point or continuous
- Help

For more information on the Vector Sketch or electromagnetic digitizers, contact:

GTCO Corporation
1055 First Street, Rockville, MD 20850
(301) 279-9550 Telex: 898471

Dealer Inquires Invited

CIRCLE 276 ON READER SERVICE CARD

dards established in the Macro Assembler manual. These standards suggest that each different section of a program (stack, coding, and data) be allotted its own segment. EXE programs may have multiple segments (each up to 64K bytes in length); this permits the assembly language programmer to fully exploit the capabilities of segmentation. It's very possible to build an EXE program that requires hundreds of kilo-bytes of memory. If a form of IBM PC memory protection is ever utilized (defining portions of RAM to be read-only and, thus, protected from inadvertent writing), segmentation is most likely to be its basis.

When I first started researching and writing this article, I had never written a program which executed as a COM program. I didn't exactly understand the rules stated in the manuals. I knew that COM programs could not have a stack segment or have relocatable references. I did not know how to meet these requirements because all of my programs had been written according to the "rules" established by the examples in the assembler manual. I didn't even know what data elements could be defined in the code segment.

With a little effort, I was able to resolve these problems. From now on, any utility programs I write will be written to execute as COM programs.■

Another article by Tom Carter, Accessing the Serial Port from COBOL, also appears in this issue.

TECH PROGRAM LISTINGS

SETUP-1

```
A>ASM
The IBM Personal Computer Assembler
Version 1.00 ©Copyright IBM Corp 1981

Source filename        [.ASM]: DPFILE
Object filename        [DPFILE.OJB]:
Source listing         [NUL.LST]: DPFILE
Cross reference        [NUL.CRF]:

Warning     Severe
Errors      Errors
0           0

A>LINK

IBM Personal Computer Linker
Version 1.10 ©Copyright IBM Corp 1982

Object Modules [.OBJ]: DPFILE
Run File [DPFILE.EXE]:
List File [NUL.MAP]:DPFILE
Libraries [.LIB]:

A>TYPE DPFILE.MAP

Start     Stop      Length    Name      Class
00000H    00188H    0189H     CODSEG    CODE
00190H    001F3H    0064H     STACKS    STACK
00200H    004C7H    02C8H     WSTORE    DATA

Program entry point at 0000:0000
```

LIST-1

```
The IBM Personal Computer Assembler 05-05-83          PAGE    1-1
DPFILE

                                 PAGE
                         TITLE   DPFILE
                         ;
                         ;       (C)COPYRIGHT 1983 - REAL TIME CONCEPTS, INC.
                         ;
                         ;       AUTHOR: THOMAS V. CARTER
                         ;
                         ;       PC DOS FILE DUMP UTILITY PROGRAM
                         ;
                         ;       ALLOCATE THE STACK WORK AREA
                         ;
0000                     STACKS  SEGMENT PARA STACK 'STACK'
                         ;
0000     32 [                    DB      50 DUP('$$')
               24 24
                     ]

                         ;
0064                     STACKS  ENDS
                         ;

The IBM Personal Computer Assembler 05-05-83          PAGE    1-2
DPFILE

                                 PAGE
                         ;
                         ;       ALLOCATE THE WORKING STORAGE SEGMENT
                         ;
0000                     WSTORE  SEGMENT PARA PUBLIC 'DATA'
                         ;
                         ;       ALLOCATE THE FILE NAME OPERATOR PROMPT
                         ;
0000  46 69 6C 65 20 6E  FILNAM  DB      'File name = '
      61 6D 65 20 3D 20
000C  24                         DB      '$'
                         ;
                         ;       ALLOCATE THE BLOCK NUMBER TEXT AND WORK AREAS
                         ;
000D  42 6C 6F 63 6B 20  BLKNUM  DB      'Block Number = '
      4E 75 6D 62 65 72
      20 3D 20
                         ;
001C  30 30 30 30        BCOUNT  DB      '0000'
0020  0A0D                       DW      0A0DH
0022  0A0D                       DW      0A0DH
0024  24                         DB      '$'
                         ;
                         ;       ALLOCATE THE FILE OPEN ERROR ERROR MESSAGE
                         ;
0025  46 69 6C 65 20 6F  OPNERM  DB      'File open error'
      70 65 6E 20 65 72
      72 6F 72
0034  0A0D                       DW      0A0DH
0036  24                         DB      '$'
                         ;
                         ;       ALLOCATE THE FILE NOT FOUND ERROR MESSAGE
                         ;
0037  46 69 6C 65 20 6E  NOFILE  DB      'File not found'
      6F 74 20 66 6F 75
      6E 64
0045  0A0D                       DW      0A0DH
0047  24                         DB      '$'
                         ;
                         ;       DEFINE CARRIAGE RETURN AND LINE FEED CHARACTERS
                         ;
0048  0A0D               CRLFLN  DW      0A0DH
004A  24                         DB      '$'
                         ;
                         ;       BINARY TO HEXADECIMAL CONVERSION TABLE
                         ;
004B  30 31 32 33 34 35  ASCIIT  DB      '0123456789ABCDEF'
      36 37 38 39 41 42
      43 44 45 46
                         ;

The IBM Personal Computer Assembler 05-05-83          PAGE    1-3
DPFILE

                                 PAGE
                         ;
                         ;       ALLOCATE FORM FEED CHARACTER
                         ;
005B  0A0D               FMFEED  DW      0A0DH
005D  0C                         DB      0CH
005E  24                         DB      '$'
                         ;
                         ;       ALLOCATE THE I/O BUFFER // 512 BYTES PER BLOCK
                         ;
005F  0200 [             FILBUF  DB      512 DUP(' ')
               20
                   ]

                         ;
                         ;       ALLOCATE THE FILESPEC INPUT STRUCTURE
                         ;
025F  0F                 INPNAM  DB      15
0260  00                         DB      0
0261     0F [                    DB      15 DUP(' ')
               20
                   ]
```

```
028E  0A0D                      DW      0A0DH
                        ;
                        ;       ALLOCATE THE HEXADECIMAL BUFFER OFFSET COUNTER
                        ;
02C0  0000              HEXCNT  DW      0
                        ;
                        ;       DEFINE BINARY TO DECIMAL DIVISORS
                        ;
02C2  03E8              BINDEC  DW      1000
02C4  0064                      DW      100
02C6  000A                      DW      10
                        ;
02C8                    WSTORE  ENDS

                                PAGE
                        ;
                        ;       PC DOS FILE DUMP UTILITY PROGRAM
                        ;
0000                    CODSEG  SEGMENT PARA PUBLIC 'CODE'
                        ;
0000                    DPFILE  PROC    FAR
                                ASSUME  CS:CODSEG               ;CS = CODE SEGMENT
                                ASSUME  DS:WSTORE               ;DS = DATA SEGMENT
                                ASSUME  SS:STACKS               ;SS = STACK SEGMENT
                        ;
                        ;       INITIALIZE THE PROGRAM ENVIRONMENT
                        ;
0000  1E                        PUSH    DS                      ;SAVE PROGRAM PREFIX SEGMENT
0001  2B C0                     SUB     AX,AX                   ;AX = ZERO // SET FOR STACK
0003  8B E8                     MOV     BP,AX                   ;BP = ZERO // STACK START
0005  89 46 00                  MOV     [BP],AX                 ;SETUP BLOCK NUMBER ZERO
0008  50                        PUSH    AX                      ;PUSH AX (ZERO) ON STACK
0009  B8 ---- R                 MOV     AX,WSTORE               ;AX = WORKING STORAGE SEGMENT
000C  8E D8                     MOV     DS,AX                   ;DS = WORKING STORAGE SEGMENT
                        ;
                        ;       DISPLAY THE FILE NAME PROMPT TO THE OPERATOR
                        ;
000E  BA 0000 R                 MOV     DX,OFFSET FILNAM        ;DX = PROMPT OFFSET ADDRESS
0011  B4 09                     MOV     AH,09H                  ;AH = 9 = DISPLAY PROMPT
0013  CD 21                     INT     21H                     ;CALL DOS FOR SERVICE
                        ;
                        ;       ACQUIRE THE OPERATOR RESPONSE // FILESPEC
                        ;
0015  BA 025F R                 MOV     DX,OFFSET INPNAM        ;DX = SEGMENT OFFSET ANSWER
0018  B4 0A                     MOV     AH,0AH                  ;AH = A = KEYBOARD INPUT
001A  CD 21                     INT     21H                     ;CALL DOS FOR SERVICE
                        ;
                        ;       SEND CARRIAGE RETURN AND LINE FEED FOR APPEARANCES
                        ;
001C  BA 0048 R                 MOV     DX,OFFSET CRLFLN        ;DX = CR/LF OFFSET ADDRESS
001F  B4 09                     MOV     AH,09H                  ;AH = 9 = DISPLAY CR AND LF
0021  CD 21                     INT     21H                     ;CALL DOS FOR SERVICE
                        ;
                        ;       PARSE THE OPERATOR SPECIFIED FILESPEC
                        ;
0023  BE 0261 R                 MOV     SI,OFFSET INPNAM+2      ;SI = FILESPEC OFFSET ADDRESS
0026  BF 005C                   MOV     DI,5CH                  ;DI = 1ST PROGRAM PREFIX FCB
0029  2A C0                     SUB     AL,AL                   ;AL = ZERO
002B  B4 29                     MOV     AH,29H                  ;PARSE THE RECEIVED FILE NAME
002D  CD 21                     INT     21H                     ;CALL DOS FOR SERVICE

                                PAGE
                        ;
                        ;       FILESPEC PARSED (1ST PROGRAM PREFIX FCB) // OPEN THE FILE
                        ;
002F  1E                        PUSH    DS                      ;SAVE THE DATA SEGMENT REGISTER
                        ;
0030  BA 005C                   MOV     DX,5CH                  ;DX = 1ST PROGRAM PREFIX FCB
0033  8C C0                     MOV     AX,ES                   ;AX = PROGRAM PREFIX SEGMENT
0035  8E D8                     MOV     DS,AX                   ;DS = PROGRAM PREFIX SEGMENT
0037  B4 0F                     MOV     AH,0FH                  ;AH = F = OPEN THE FILE
0039  CD 21                     INT     21H                     ;CALL DOS FOR SERVICE
                        ;
003B  1F                        POP     DS                      ;RESTORE DATA SEGMENT REGISTER
                        ;
003C  3C 00                     CMP     AL,0                    ;AL = OPEN FILE RETURN STATUS
003E  74 08                     JE      DPF010                  ;BRANCH IF SUCCESSFUL OPEN
                        ;
                        ;       FILE OPEN ERROR - NOTIFY THE OPERATOR AND TERMINATE
                        ;
0040  BA 0025 R                 MOV     DX,OFFSET OPNERM        ;DX = OPEN ERROR MESSAGE OFFSET
0043  B4 09                     MOV     AH,09H                  ;AH = 9 = DISPLAY ERROR MESSAGE
0045  CD 21                     INT     21H                     ;CALL DOS FOR SERVICE
0047  CB                        RET                             ;TERMINATE PROGRAM EXECUTION
                        ;
                        ;       SEND CARRIAGE RETURN AND LINE FEED FOR APPEARANCES
                        ;
0048                    DPF010:
0048  BA 0048 R                 MOV     DX,OFFSET CRLFLN        ;DX = CR/LF OFFSET ADDRESS
004B  B4 09                     MOV     AH,09H                  ;AH = 9 = DISPLAY CR AND LF
004D  CD 21                     INT     21H                     ;CALL DOS FOR SERVICE
                        ;
                        ;       SETUP THE DATA TRANSFER ADDRESS (DTA)
                        ;
004F  BA 005F R                 MOV     DX,OFFSET FILBUF        ;DX = I/O BUFFER OFFSET ADDRESS
0052  B4 1A                     MOV     AH,1AH                  ;AH = 1A = SETUP DTA
0054  CD 21                     INT     21H                     ;CALL DOS FOR SERVICE
                        ;
                        ;       INITIALIZE THE 1ST PROGRAM PREFIX FILE CONTROL BLOCK
                        ;
0056  BF 005C                   MOV     DI,5CH                  ;DI = FCB OFFSET ADDRESS
0059  B8 0200                   MOV     AX,512                  ;AX = 512 = RECORD SIZE
005C  26: 89 45 0E              MOV     ES:[DI+14],AX           ;SETUP LOGICAL RECORD SIZE
0060  2B C0                     SUB     AX,AX                   ;AX = ZERO
0062  26: 89 45 20              MOV     ES:[DI+32],AX           ;SETUP RANDOM RECORD NUMBER
0066  26: 89 45 22              MOV     ES:[DI+34],AX           ;SETUP RANDOM RECORD NUMBER
006A  26: 88 45 24              MOV     ES:[DI+36],AL           ;SETUP RANDOM RECORD NUMBER
```

```
                                PAGE
                        ;
                        ;       SETUP TO CLEAR THE FILE I/O BUFFER
                        ;
006E                    DPF020:
006E  B9 0100                   MOV     CX,256                  ;CX = LOOP CONTROL COUNTER
0071  BF 005F R                 MOV     DI,OFFSET FILBUF        ;DI = I/O BUFFER OFFSET ADDRESS
0074  2B C0                     SUB     AX,AX                   ;AX = ZEROS
                        ;
                        ;       SET THE CONTENTS OF THE I/O BUFFER TO BINARY ZEROS
                        ;
0076                    DPF030:
0076  89 05                     MOV     [DI],AX                 ;SET I/O BUFFER TO BINARY ZEROS
0078  83 C7 02                  ADD     DI,2                    ;INCREMENT DI TO NEXT WORD
007B  E2 F9                     LOOP    DPF030                  ;LOOP FOR ALL BUFFER WORDS
                        ;
                        ;       READ EACH 512 BYTE BLOCK OF THE FILE // RANDOM BLOCK READ
                        ;
007D  1E                        PUSH    DS                      ;SAVE DATA SEGMENT REGISTER
007E  8C C0                     MOV     AX,ES                   ;AX = PROGRAM PREFIX SEGMENT
0080  8E D8                     MOV     DS,AX                   ;DS = PROGRAM PREFIX SEGMENT
0082  BA 005C                   MOV     DX,5CH                  ;DX = 1ST PROGRAM PREFIX FCB
0085  B9 0001                   MOV     CX,1                    ;CX = NUMBER OF RECORDS
0088  B4 27                     MOV     AH,27H                  ;AH = 27 = RANDOM BLOCK READ
008A  CD 21                     INT     21H                     ;CALL DOS FOR SERVICE
008C  1F                        POP     DS                      ;RESTORE DATA SEGMENT REGISTER
                        ;
008D  3C 00                     CMP     AL,0                    ;VERIFY DISK READ COMPLETION
008F  74 06                     JE      DPF040                  ;BRANCH IF SUCCESSFUL READ
0091  83 F9 00                  CMP     CX,0                    ;CHECK FOR PARTIAL BLOCK
0094  75 01                     JNE     DPF040                  ;BRANCH IF DATA IN BUFFER
                        ;
0096  CB                        RET                             ;TERMINATE PROGRAM EXECUTION
                        ;
                        ;       CONVERT DATA TO DISPLAYABLE ASCII AND DISPLAY
                        ;
0097                    DPF040:
0097  BF 001C R                 MOV     DI,OFFSET BCOUNT        ;DI = BLOCK COUNT ADDRESS
009A  8B 46 00                  MOV     AX,[BP]                 ;AX = CURRENT BLOCK COUNT
009D  40                        INC     AX                      ;INCREMENT BLOCK COUNT
009E  89 46 00                  MOV     [BP],AX                 ;SAVE NEW BLOCK COUNT
00A1  48                        DEC     AX                      ;DECREMENT BLOCK COUNT
00A2  E8 0160 R                 CALL    BDCONV                  ;BINARY TO DECIMAL CONVERSION
00A5  BA 000D R                 MOV     DX,OFFSET BLKNUM        ;DX = BLOCK NUMBER MESSAGE
00A8  B4 09                     MOV     AH,09H                  ;AH = 9 = DISPLAY BLOCK NUMBER
00AA  CD 21                     INT     21H                     ;CALL DOS FOR SERVICE
00AC  BE 005F R                 MOV     SI,OFFSET FILBUF        ;SI = FILE I/O BUFFER ADDRESS
00AF  BF 027C R                 MOV     DI,OFFSET DPLINE+12     ;DI = HEXADECIMAL DISPLAY AREA
00B2  BD 02AE R                 MOV     BP,OFFSET DPLINE+62     ;BP = ASCII DISPLAY AREA
00B5  B9 0020                   MOV     CX,32                   ;CX = NUMBER OF 16 BYTE LINES
00B8  51                        PUSH    CX                      ;SAVE CX REGISTER ON THE STACK
00B9  B9 0010                   MOV     CX,16                   ;CONVERSION LOOP COUNTER

                                PAGE
                        ;
                        ;       CONVERT A BYTE OF INPUT DATA TO HEXADECIMAL
                        ;
00BC                    DPF050:
00BC  83 F9 08                  CMP     CX,8                    ;HALF OF LINE CONVERTED?
00BF  75 06                     JNE     DPF060                  ;BRANCH TO CONTINUE - NOT HALF
00C1  4F                        DEC     DI                      ;DECREMENT LINE POSITION INDEX
00C2  B0 2D                     MOV     AL,'-'                  ;MOVE A DASH TO SEPARATE HALFS
00C4  88 05                     MOV     [DI],AL                 ;MOVE A DASH TO SEPARATE HALFS
00C6  47                        INC     DI                      ;INCREMENT LINE POSITION INDEX
                        ;
                        ;       SAVE CX ON THE STACK AND PROCESS NEXT BYTE
                        ;
00C7                    DPF060:
00C7  51                        PUSH    CX                      ;SAVE CONVERSION LOOP COUNTER
00C8  B1 04                     MOV     CL,4                    ;CL = 4 = SHIFT COUNTER
00CA  8A 04                     MOV     AL,[SI]                 ;AL = FILE BUFFER CHARACTER
00CC  8A E0                     MOV     AH,AL                   ;AH = FILE BUFFER CHARACTER
00CE  80 FC 20                  CMP     AH,20H                  ;COMPARE AH AN ASCII " "
00D1  7C 05                     JL      DPF070                  ;BRANCH IF UNDISPLAYABLE
00D3  80 FC 7E                  CMP     AH,7EH                  ;COMPARE AH TO ASCII "~"
00D6  7E 02                     JNG     DPF080                  ;BRANCH IF DISPLAYABLE
                        ;
                        ;       SET AH TO AN ASCII PERIOD // NON-DISPLAYABLE CHARACTER
                        ;
00D8                    DPF070:
00D8  B4 2E                     MOV     AH,"."                  ;AH = AN ASCII PERIOD
                        ;
                        ;       AH CONTAINS THE DISPLAYABLE ASCII CHARACTER
                        ;
00DA                    DPF080:
00DA  3E: 88 66 00              MOV     DS:[BP],AH              ;MOVE CHARACTER TO ASCII AREA
00DE  2A E4                     SUB     AH,AH                   ;AH = ZEROS // SET FOR CONVERT
00E0  D3 E0                     SHL     AX,CL                   ;POSITION UPPER CL BITS OF AL
00E2  D2 E8                     SHR     AL,CL                   ;POSITION LOWER CL BITS OF AL
00E4  86 C4                     XCHG    AL,AH                   ;EXCHANGE CONTENTS OF AL AND AH
00E6  50                        PUSH    AX                      ;SAVE CONTENTS OF AX ON STACK
00E7  BB 004B R                 MOV     BX,OFFSET ASCIIT        ;BX = ASCII TABLE ADDRESS
00EA  D7                        XLAT    ASCIIT                  ;CONVERT CHARACTER TO HEX
00EB  88 05                     MOV     [DI],AL                 ;MOVE CHARACTER TO DISPLAY LINE
00ED  47                        INC     DI                      ;INCREMENT STORAGE INDEX
00EE  58                        POP     AX                      ;RESTORE ORIGINAL AX CONTENTS
00EF  B1 08                     MOV     CL,8                    ;CL = 8 = SHIFT COUNTER
00F1  D3 E8                     SHR     AX,CL                   ;POSITION UPPER 8 BITS
00F3  D7                        XLAT    ASCIIT                  ;CONVERT CHARACTER TO HEX
00F4  88 05                     MOV     [DI],AL                 ;MOVE CHARACTER TO DISPLAY LINE
00F6  83 C7 02                  ADD     DI,2                    ;INCREMENT DI FOR NEXT BYTE
00F9  46                        INC     SI                      ;INCREMENT SI FOR NEXT BYTE
00FA  45                        INC     BP                      ;INCREMENT BP FOR NEXT BYTE
00FB  59                        POP     CX                      ;RESTORE LOOP CONTROL COUNTER
00FC  E2 BE                     LOOP    DPF050                  ;BRANCH TO CONVERT FULL LINE

                                PAGE
                        ;
                        ;       CONVERT AND SETUP RELATIVE BUFFER OFFSET
                        ;
00FE  A1 02C0 R                 MOV     AX,HEXCNT               ;AX = CURRENT BUFFER OFFSET
0101  BF 0276 R                 MOV     DI,OFFSET DPLINE+6      ;DI = DISPLAY LINE OFFSET
0104  56                        PUSH    SI                      ;SAVE BUFFER INDEX ON STACK
0105  E8 0160 R                 CALL    BDCONV                  ;BINARY TO DECIMAL CONVERSION
0108  5E                        POP     SI                      ;RESTORE BUFFER INDEX
0109  BF 0270 R                 MOV     DI,OFFSET DPLINE        ;DI = DISPLAY LINE OFFSET
010C  A1 02C0 R                 MOV     AX,HEXCNT               ;AX = CURRENT BUFFER OFFSET
010F  B9 0004                   MOV     CX,4                    ;CX = LOOP COUNTER (4 BYTES)
```

```
                        ;       CONVERT HEXADECIMAL TO ASCII // BUFFER OFFSET
                        ;
 0112                   DPF090:
 0112  51                       PUSH    CX                      ;SAVE LOOP COUNTER ON STACK
 0113  B1 04                    MOV     CL,4                    ;CL = 4 = SHIFT COUNTER
 0115  50                       PUSH    AX                      ;SAVE CURRENT BUFFER OFFSET
 0116  25 F000                  AND     AX,0F000H               ;ISOLATE UPPER NIBBLE // 4 BITS
 0119  86 C4                    XCHG    AL,AH                   ;EXCHANGE VALUES IN AL AND AH
 011B  D2 E8                    SHR     AL,CL                   ;POSITION 4-BIT NIBBLE
 011D  D7                       XLAT    ASCIIT                  ;CONVERT TO HEXADECIMAL BYTE
 011E  88 05                    MOV     [DI],AL                 ;STORE CONVERTED BYTE // HEX
 0120  47                       INC     DI                      ;INCREMENT BYTE STORAGE INDEX
 0121  58                       POP     AX                      ;RESTORE CONTENTS TO AX
 0122  D3 E0                    SHL     AX,CL                   ;POSITION NEXT 4-BIT NIBBLE
 0124  59                       POP     CX                      ;RESTORE THE LOOP COUNTER
 0125  E2 EB                    LOOP    DPF090                  ;LOOP TO CONVERT ALL NIBBLES
                        ;
                        ;       DISPLAY THE LAST LINE CONVERTED
                        ;
 0127  83 06 02C0 R 10          ADD     HEXCNT,16               ;UPDATE HEX OFFSET COUNTER
 012C  BB 0270 R                MOV     BX,OFFSET DPLINE        ;BX = HEXADECIMAL DISPLAY AREA
 012F  B9 0050                  MOV     CX,80                   ;CX = DISPLAY LINE BYTE COUNT
                        ;
                        ;       DISPLAY A CONVERTED BYTE OF DATA
                        ;
 0132                   DPF100:
 0132  8A 17                    MOV     DL,[BX]                 ;DL = CONVERTED DATA BYTE
 0134  43                       INC     BX                      ;INCREMENT BX TO NEXT BYTE
 0135  B4 02                    MOV     AH,02H                  ;AH = 2 = DISPLAY CHARACTER
 0137  CD 21                    INT     21H                     ;CALL DOS FOR SERVICE
 0139  E2 F7                    LOOP    DPF100                  ;BRANCH TO DISPLAY NEXT BYTE
 013B  59                       POP     CX                      ;RESTORE BLOCK COUNTER
 013C  E2 12                    LOOP    DPF110                  ;BRANCH TO PRINT NEXT LINE
 013E  C7 06 02C0 R 0000        MOV     HEXCNT,0                ;RESET THE HEX OFFSET COUNTER
 0144  BA 005B R                MOV     DX,OFFSET FMFEED        ;DX = CR, LF, AND FF BUFFER
 0147  B4 09                    MOV     AH,09H                  ;AH = 9 = WRITE DATA TO DISPLAY
 0149  CD 21                    INT     21H                     ;CALL DOS FOR SERVICE
 014B  2B ED                    SUB     BP,BP                   ;BP = ZERO - FOR BLOCK NUMBER
 014D  E9 006E R                JMP     DPF020                  ;BRANCH TO READ NEXT BLOCK
                        ;
                                PAGE
                        ;
                        ;       CONVERT AND DISPLAY NEXT 16 BYTES OF DATA
                        ;
 0150                   DPF110:
 0150  51                       PUSH    CX                      ;SAVE CONTENTS OF LINE COUNTER
 0151  B8 2020                  MOV     AX,2020H                ;AX = 2 ASCII SPACES
 0154  B9 0027                  MOV     CX,39                   ;CX = NUMBER OF WORDS TO SET
 0157  BF 0270 R                MOV     DI,OFFSET DPLINE        ;DI = DISPLAY LINE OFFSET
                        ;
                        ;       MOVE ASCII SPACES TO THE DISPLAY LINE BUFFER
                        ;
 015A                   DPF120:
 015A  89 05                    MOV     [DI],AX                 ;MOVE SPACES TO DISPLAY LINE
 015C  83 C7 02                 ADD     DI,2                    ;INCREMENT DI BY WORDS
 015F  E2 F9                    LOOP    DPF120                  ;BRANCH TO SET ALL WORDS
                        ;
 0161  BF 027C R                MOV     DI,OFFSET DPLINE+12     ;DI = HEXADECIMAL DISPLAY AREA
 0164  BD 02AE R                MOV     BP,OFFSET DPLINE+62     ;BP = ASCII DISPLAY AREA
 0167  B9 0010                  MOV     CX,16                   ;RESET LINE CONVERSION COUNTER
 016A  E9 008C R                JMP     DPF050                  ;BRANCH TO CONVERT NEXT LINE
                        ;
 016D                   DPFILE  ENDP
                        ;
                                PAGE
                        ;
                        ;       BINARY TO DECIMAL (ASCII) CONVERSION
                        ;
 016D                   BDCONV  PROC    NEAR
 016D  B9 0003                  MOV     CX,3                    ;CX = LOOP COUNTER
 0170  2B F6                    SUB     SI,SI                   ;SI = ZERO // FOR INDEX
                        ;
                        ;       BINARY TO DECIMAL CONVERSION LOOP
                        ;
 0172                   BDC010:
 0172  2B D2                    SUB     DX,DX                   ;DX = ZERO // FOR DIVISION
 0174  F7 B4 02C2 R             DIV     BINDEC[SI]              ;DIVIDE FOR DECIMAL DIGIT
 0178  04 30                    ADD     AL,30H                  ;ADD 30H FOR ASCII NUMBER
 017A  88 05                    MOV     [DI],AL                 ;STORE ASCII NUMERIC DIGIT
 017C  47                       INC     DI                      ;INCREMENT FOR NEXT NUMBER
 017D  83 C6 02                 ADD     SI,2                    ;INCREMENT SI FOR NEXT NUMBER
 0180  8B C2                    MOV     AX,DX                   ;RESTORE REMAINDER TO AX
 0182  E2 EE                    LOOP    BDC010                  ;BRANCH TO CONVERT ALL DATA
                        ;
 0184  04 30                    ADD     AL,30H                  ;ADD 30H FOR ASCII
 0186  88 05                    MOV     [DI],AL                 ;STORE ASCII NUMERIC DIGIT
 0188  C3                       RET                             ;RETURN TO CALLER
                        ;
 0189                   BDCONV  ENDP
                        ;
 0189                   CODSEG  ENDS
                                END     DPFILE


 The IBM Personal Computer Assembler 05-05-83           PAGE    Symbols-1
 DPFILE


 Segments and groups:

                 N a m e                Size    align   combine class

 CODSEG . . . . . . . . . . . . .       0189    PARA    PUBLIC  'CODE'
 STACKS . . . . . . . . . . . . .       0064    PARA    STACK   'STACK'
 WSTORE . . . . . . . . . . . . .       02C8    PARA    PUBLIC  'DATA'

 Symbols:

                 N a m e                Type    Value   Attr

 ASCIIT . . . . . . . . . . . . .       L BYTE  0048    WSTORE
 BCOUNT . . . . . . . . . . . . .       L BYTE  001C    WSTORE
 BDC010 . . . . . . . . . . . . .       L NEAR  0172    CODSEG
 BDCONV . . . . . . . . . . . . .       N PROC  016D    CODSEG  Length =001C
 BINDEC . . . . . . . . . . . . .       L WORD  02C2    WSTORE
 BLKNUM . . . . . . . . . . . . .       L BYTE  000D    WSTORE
 CRLFLN . . . . . . . . . . . . .       L WORD  0048    WSTORE
 DPF010 . . . . . . . . . . . . .       L NEAR  0048    CODSEG
 DPF020 . . . . . . . . . . . . .       L NEAR  006E    CODSEG
 DPF030 . . . . . . . . . . . . .       L NEAR  0076    CODSEG
 DPF040 . . . . . . . . . . . . .       L NEAR  0097    CODSEG
 DPF050 . . . . . . . . . . . . .       L NEAR  00BC    CODSEG
 DPF060 . . . . . . . . . . . . .       L NEAR  00C7    CODSEG
 DPF070 . . . . . . . . . . . . .       L NEAR  00D8    CODSEG
 DPF080 . . . . . . . . . . . . .       L NEAR  00DA    CODSEG
 DPF090 . . . . . . . . . . . . .       L NEAR  0112    CODSEG
 DPF100 . . . . . . . . . . . . .       L NEAR  0132    CODSEG
 DPF110 . . . . . . . . . . . . .       L NEAR  0150    CODSEG
 DPF120 . . . . . . . . . . . . .       L NEAR  015A    CODSEG
 DPFILE . . . . . . . . . . . . .       F PROC  0000    CODSEG  Length =016D
 DPLINE . . . . . . . . . . . . .       L BYTE  0270    WSTORE  Length =004E
 FILBUF . . . . . . . . . . . . .       L BYTE  005F    WSTORE  Length =0200
 FILNAM . . . . . . . . . . . . .       L BYTE  0000    WSTORE
 FMFEED . . . . . . . . . . . . .       L WORD  005B    WSTORE
 HEXCNT . . . . . . . . . . . . .       L WORD  02C0    WSTORE
 INPNAM . . . . . . . . . . . . .       L BYTE  025F    WSTORE
 NOFILE . . . . . . . . . . . . .       L BYTE  0037    WSTORE
 OPNERM . . . . . . . . . . . . .       L BYTE  0025    WSTORE

 Warning Severe
 Errors  Errors
 0       0
```

DUMP-1

```
File name = dpfile.exe

Block Number = 0000

0000  0000  4D 5A C8 00 04 00 01 00-20 00 00 00 00 00 19 00   MZ...... .......
0010  0016  64 00 17 31 00 00 00 00-1C 00 00 00 0A 00 00 00   d..1............
0020  0032  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0030  0048  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0040  0064  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0050  0080  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0060  0096  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0070  0112  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0080  0128  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0090  0144  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
00A0  0160  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
00B0  0176  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
00C0  0192  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
00D0  0208  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
00E0  0224  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
00F0  0240  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0100  0256  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0110  0272  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0120  0288  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0130  0304  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0140  0320  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0150  0336  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0160  0352  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0170  0368  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0180  0384  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0190  0400  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
01A0  0416  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
01B0  0432  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
01C0  0448  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
01D0  0464  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
01E0  0480  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
01F0  0496  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................


Block Number = 0001

0000  0000  1E 2B C0 8B E8 89 46 00-50 B8 20 00 8E D8 BA 00   .+....F.P. .....
0010  0016  00 B4 09 CD 21 BA 5F 02-B4 0A CD 21 BA 48 00 B4   ....!._....!.H..
0020  0032  09 CD 21 BE 61 02 BF 5C-00 2A C0 B4 29 CD 21 1E   ..!.a..\.*..).!.
0030  0048  BA 5C 00 8C C0 8E D8 B4-0F CD 21 1F 3C 00 74 08   .\.......!.<.t.
0040  0064  BA 25 00 B4 09 CD 21 CB-BA 48 00 B4 09 CD 21 BA   .%....!..H....!.
0050  0080  5F 00 B4 1A CD 21 BF 5C-00 B8 00 02 26 89 45 0E   _....!.\....&.E.
0060  0096  2B C0 26 89 45 20 26 89-45 22 26 88 45 24 B9 00   +.&.E &.E"&.E$..
0070  0112  01 BF 5F 00 2B C0 89 05-83 C7 02 E2 F9 1E 8C C0   .._.+...........
0080  0128  8E D8 BA 5C 00 B9 01 00-B4 27 CD 21 1F 3C 00 74   ...\....'.!.<.t
0090  0144  06 83 F9 00 75 01 CB BF-1C 00 8B 46 00 40 89 46   ....u......F.@.F
00A0  0160  00 48 E8 C8 00 BA 0D 00-B4 09 CD 21 BE 5F 00 BF   .H.........!._..
00B0  0176  7C 02 BD AE 02 B9 20 00-51 B9 10 00 83 F9 08 75   |..... .Q......u
00C0  0192  06 4F B0 2D 88 05 47 51-B1 04 8A 04 8A E0 80 FC   .O.-..GQ........
00D0  0208  20 7C 05 80 FC 7E 7E 02-B4 2E 3E 88 66 00 2A E4    |...~~...>.f.*.


DUMP1 -- printed on 07-06-1983 at 15:44:03 -- Page 2


00E0  0224  D3 E0 D2 E8 86 C4 50 BB-4B 00 D7 88 05 47 58 B1   ......P.K....GX.
00F0  0240  08 D3 E8 D7 88 05 83 C7-02 46 45 59 E2 BE A1 C0   .........FEY....
0100  0256  02 BF 76 02 56 E8 65 00-5E BF 70 02 A1 C0 02 B9   ..v.V.e.^.p.....
0110  0272  04 00 51 B1 04 50 25 00-F0 86 C4 D2 E8 D7 88 05   ..Q..P%.........
0120  0288  47 58 D3 E0 59 E2 EB 83-06 C0 02 10 BB 70 02 B9   GX..Y........p..
0130  0304  50 00 8A 17 43 B4 02 CD-21 E2 F7 59 E2 12 C7 06   P...C...!..Y....
0140  0320  C0 02 00 00 BA 5B 00 B4-09 CD 21 2B ED E9 1E FF   .....[....!+....
0150  0336  51 B8 20 20 B9 27 00 BF-70 02 89 05 83 C7 02 E2   Q.  .'..p.......
0160  0352  F9 BF 7C 02 BD AE 02 B9-10 00 E9 4F FF B9 03 00   ..|........O....
0170  0368  2B F6 2B D2 F7 B4 C2 02-04 30 88 05 47 83 C6 02   +.+......0..G...
0180  0384  8B C2 E2 EE 04 30 88 05-C3 00 00 00 00 00 00 00   .....0..........
0190  0400  24 24 24 24 24 24 24 24-24 24 24 24 24 24 24 24   $$$$$$$$$$$$$$$$
```

```
01A0  0416  24 24 24 24 24 24 24 24-24 24 24 24 24 24 24 24   $$$$$$$$$$$$$$$$
01B0  0432  24 24 24 24 24 24 24 24-24 24 24 24 24 24 24 24   $$$$$$$$$$$$$$$$
01C0  0448  24 24 24 24 24 24 24 24-24 24 24 24 24 24 24 24   $$$$$$$$$$$$$$$$
01D0  0464  24 24 24 24 24 24 24 24-24 24 24 24 24 24 24 24   $$$$$$$$$$$$$$$$
01E0  0480  24 24 24 24 24 24 24 24-24 24 24 24 24 24 24 24   $$$$$$$$$$$$$$$$
01F0  0496  24 24 24 24 00 00 00 00-00 00 00 00 00 00 00 00   $$$$............

Block Number = 0002

0000  0000  46 69 6C 65 20 6E 61 6D-65 20 3D 20 24 42 6C 6F   File name = $Blo
0010  0016  63 6B 20 4E 75 6D 62 65-72 20 3D 20 30 30 30 30   ck Number = 0000
0020  0032  0D 0A 0D 0A 24 46 69 6C-65 20 6F 70 65 6E 20 65   ....$File open e
0030  0048  72 72 6F 72 0D 0A 24 46-69 6C 65 20 6E 6F 74 20   rror..$File not
0040  0064  66 6F 75 6E 64 0D 0A 24-0D 0A 24 30 31 32 33 34   found..$..$01234
0050  0080  35 36 37 38 39 41 42 43-44 45 46 0D 0A 0C 24 20   56789ABCDEF...$
0060  0096  20 20 20 20 20 20 20 20-20 20 20 20 20 20 20 20

Block Number = 0003

0000  0000  20 20 20 20 20 20 20 20-20 20 20 20 20 20 20 20
0010  0016  20 20 20 20 20 20 20 20-20 20 20 20 20 20 20 20
0020  0032  20 20 20 20 20 20 20 20-20 20 20 20 20 20 20 20
0030  0048  20 20 20 20 20 20 20 20-20 20 20 20 20 20 20 20
0040  0064  20 20 20 20 20 20 20 20-20 20 20 20 20 20 20 20
0050  0080  20 20 20 20 20 20 20 20-20 20 20 20 20 20 20 0F
0060  0096  00 20 20 20 20 20 20 20-20 20 20 20 20 20 20 20
0070  0112  20 20 20 20 20 20 20 20-20 20 20 20 20 20 20 20
0080  0128  20 20 20 20 20 20 20 20-20 20 20 20 20 20 20 20
0090  0144  20 20 20 20 20 20 20 20-20 20 20 20 20 20 20 20
00A0  0160  20 20 20 20 20 20 20 20-20 20 20 20 20 20 20 20
00B0  0176  20 20 20 20 20 20 20 20-20 20 20 20 20 20 0D 0A                 ..
00C0  0192  00 00 E8 03 64 00 0A 00-00 00 00 00 00 00 00 00   ....d...........
```

SETUP-2

A)ASM
The IBM Personal Computer Assembler
Version 1.00 ©Copyright IBM Corp 1981

Source filename [.ASM: DPFCOM
Object filename [DPFCOM.OBJ]:
Source listing [NUL.LST]: DPFCOM
Cross reference [NUL.CRF]:

Warning Errors	Severe Errors
0	0

A)LINK

IBM Personal Computer Linker
Version 1.10 ©Copyright IBM Corp 1982

Object Modules [.OBJ]: DPFCOM
Run File [DPFCOM.EXE]:
List File [NUL.MAP]:DPFCOM
Libraries [.LIB]:
Warning: No STACK segment

There was 1 error detected.

A)EXE2BIN DPFCOM.EXE

A)RENAME DPFCOM.BIN DPFCOM.COM

A)TYPE DPFCOM.MAP

Warning: No STACK segment

Start	Stop	Length	Name	Class
00000H	00543H	0544H	CODSEG	CODE

Program entry point at 0000:0100

LIST-2

```
        PAGE 60,132
        PAGE
TITLE   DPFILE
;
;       (C)COPYRIGHT 1983 - REAL TIME CONCEPTS, INC.
;
;       AUTHOR: THOMAS V. CARTER
;
;       PC DOS FILE DUMP UTILITY PROGRAM
;
CODSEG SEGMENT PARA PUBLIC 'CODE'
;
;       START ADDRESS IS AT OFFSET 100 HEX
;
        ORG     100H                    ;SET IC TO 100 HEX
;
DPFILE  PROC    FAR
        ASSUME  CS:CODSEG               ;CS = CODE SGEMENT
        ASSUME  DS:CODSEG               ;DS = CODE SEGMENT
        ASSUME  SS:CODSEG               ;SS = CODE SEGMENT
        ASSUME  ES:CODSEG               ;ES = CODE SEGMENT
;
        JMP     SSTART                  ;BRANCH TO TRUE START ADDRESS
;
        PAGE
;
;       ALLOCATE THE WORKING STORAGE SEGMENT
;
;       ALLOCATE THE FILE NAME OPERATOR PROMPT
;
FILNAM  DB      'File name = '
        DB      '$'
;
;       ALLOCATE THE BLOCK NUMBER TEXT AND WORK AREAS
;
BLKNUM  DB      'Block Number = '
;
BCOUNT  DB      '0000'
        DW      0A0DH
        DW      0A0DH
        DB      '$'
;
;       ALLOCATE THE FILE OPEN ERROR ERROR MESSAGE
;
OPNERM  DB      'File open error'
        DW      0A0DH
        DB      '$'
;
;       ALLOCATE THE FILE NOT FOUND ERROR MESSAGE
;
NOFILE  DB      'File not found'
        DW      0A0DH
        DB      '$'
;
;
;       DEFINE CARRIAGE RETURN AND LINE FEED CHARACTERS
;
CRLFLN  DW      0A0DH
        DB      '$'
;
;       BINARY TO HEXADECIMAL CONVERSION TABLE
;
ASCIIT  DB      '0123456789ABCDEF'
;
        PAGE
;
;       ALLOCATE FORM FEED CHARACTER
;
FMFEED  DW      0A0DH
        DB      0CH
        DB      '$'
;
;       ALLOCATE THE I/O BUFFER // 512 BYTES PER BLOCK
;
FILBUF  DB      512 DUP(' ')
;
;       ALLOCATE THE FILESPEC INPUT STRUCTURE
;
INPNAM  DB      15
        DB      0
        DB      15 DUP(' ')
;
;       ALLOCATE THE FILE DUMP PRINT LINE
;
OPLINE  DB      78 DUP(' ')
        DW      0A0DH
;
;       ALLOCATE THE HEXADECIMAL BUFFER OFFSET COUNTER
;
HEXCNT  DW      0
;
;       DEFINE BINARY TO DECIMAL DIVISORS
;
BINDEC  DW      1000
        DW      100
        DW      10
;
        PAGE
;
;       PC DOS FILE DUMP UTILITY PROGRAM
;
;       INITIALIZE THE PROGRAM ENVIRONMENT
;
SSTART:
        SUB     AX,AX                   ;AX = ZERO // SET FOR STACK
        MOV     BP,80H                  ;BP = 80H // TOP OF STACK
        MOV     [BP],AX                 ;SETUP BLOCK NUMBER ZERO
        MOV     SP,100H                 ;SP = 100H = END OF STACK
;
;       DISPLAY THE FILE NAME PROMPT TO THE OPERATOR
;
        MOV     DX,OFFSET FILNAM        ;DX = PROMPT OFFSET ADDRESS
        MOV     AH,09H                  ;AH = 9 = DISPLAY PROMPT
        INT     21H                     ;CALL DOS FOR SERVICE
```

```
;
;       ACQUIRE THE OPERATOR RESPONSE // FILESPEC
;
        MOV     DX,OFFSET INPNAM        ;DX = SEGMENT OFFSET ANSWER
        MOV     AH,0AH                  ;AH = A = KEYBOARD INPUT
        INT     21H                     ;CALL DOS FOR SERVICE
;
;       SEND CARRIAGE RETURN AND LINE FEED FOR APPEARANCES
;
        MOV     DX,OFFSET CRLFLN        ;DX = CR/LF OFFSET ADDRESS
        MOV     AH,09H                  ;AH = 9 = DISPLAY CR AND LF
        INT     21H                     ;CALL DOS FOR SERVICE
;
;       PARSE THE OPERATOR SPECIFIED FILESPEC
;
        MOV     SI,OFFSET INPNAM+2      ;SI = FILESPEC OFFSET ADDRESS
        MOV     DI,5CH                  ;DI = 1ST PROGRAM PREFIX FCB
        SUB     AL,AL                   ;AL = ZERO
        MOV     AH,29H                  ;PARSE THE RECEIVED FILE NAME
        INT     21H                     ;CALL DOS FOR SERVICE
;
        PAGE
;
;       FILESPEC PARSED (1ST PROGRAM PREFIX FCB) // OPEN THE FILE
;
        MOV     DX,5CH                  ;DX = 1ST PROGRAM PREFIX FCB
        MOV     AH,0FH                  ;AH = F = OPEN THE FILE
        INT     21H                     ;CALL DOS FOR SERVICE
;
        CMP     AL,0                    ;AL = OPEN FILE RETURN STATUS
        JE      DPF010                  ;BRANCH IF SUCCESSFUL OPEN
;
;       FILE OPEN ERROR - NOTIFY THE OPERATOR AND TERMINATE
;
        MOV     DX,OFFSET OPNERM        ;DX = OPEN ERROR MESSAGE OFFSET
        MOV     AH,09H                  ;AH = 9 = DISPLAY ERROR MESSAGE
        INT     21H                     ;CALL DOS FOR SERVICE
;
        INT     20H                     ;TERMINATE PROGRAM EXECUTION
;
;       SEND CARRIAGE RETURN AND LINE FEED FOR APPEARANCES
;
DPF010:
        MOV     DX,OFFSET CRLFLN        ;DX = CR/LF OFFSET ADDRESS
        MOV     AH,09H                  ;AH = 9 = DISPLAY CR AND LF
        INT     21H                     ;CALL DOS FOR SERVICE
;
;       SETUP THE DATA TRANSFER ADDRESS (DTA)
;
        MOV     DX,OFFSET FILBUF        ;DX = I/O BUFFER OFFSET ADDRESS
        MOV     AH,1AH                  ;AH = 1A = SETUP DTA
        INT     21H                     ;CALL DOS FOR SERVICE
;
;       INITIALIZE THE 1ST PROGRAM PREFIX FILE CONTROL BLOCK
;
        MOV     DI,5CH                  ;DI = FCB OFFSET ADDRESS
        MOV     AX,512                  ;AX = 512 = RECORD SIZE
        MOV     [DI+14],AX              ;SETUP LOGICAL RECORD SIZE
        SUB     AX,AX                   ;AX = ZERO
        MOV     [DI+32],AX              ;SETUP RANDOM RECORD NUMBER
        MOV     [DI+34],AX              ;SETUP RANDOM RECORD NUMBER
        MOV     [DI+36],AL              ;SETUP RANDOM RECORD NUMBER
;
        PAGE
;
;       SETUP TO CLEAR THE FILE I/O BUFFER
;
DPF020:
        MOV     CX,256                  ;CX = LOOP CONTROL COUNTER
        MOV     DI,OFFSET FILBUF        ;DI = I/O BUFFER OFFSET ADDRESS
        SUB     AX,AX                   ;AX = ZEROS
;
;       SET THE CONTENTS OF THE I/O BUFFER TO BINARY ZEROS
;
DPF030:
        MOV     [DI],AX                 ;SET I/O BUFFER TO BINARY ZEROS
        ADD     DI,2                    ;INCREMENT DI TO NEXT WORD
        LOOP    DPF030                  ;LOOP FOR ALL BUFFER WORDS
;
;       READ EACH 512 BYTE BLOCK OF THE FILE // RANDOM BLOCK READ
;
        MOV     DX,5CH                  ;DX = 1ST PROGRAM PREFIX FCB
        MOV     CX,1                    ;CX = NUMBER OF RECORDS
        MOV     AH,27H                  ;AH = 27 = RANDOM BLOCK READ
        INT     21H                     ;CALL DOS FOR SERVICE
;
        CMP     AL,0                    ;VERIFY DISK READ COMPLETION
        JE      DPF040                  ;BRANCH IF SUCCESSFUL READ
        CMP     CX,0                    ;CHECK FOR PARTIAL BLOCK
        JNE     DPF040                  ;BRANCH IF DATA IN BUFFER
;
        INT     20H                     ;TERMINATE PROGRAM EXECUTION
;
;       CONVERT DATA TO DISPLAYABLE ASCII AND DISPLAY
;
DPF040:
        MOV     DI,OFFSET BCOUNT        ;DI = BLOCK COUNT ADDRESS
        MOV     AX,[BP]                 ;AX = CURRENT BLOCK COUNT
        INC     AX                      ;INCREMENT BLOCK COUNT
        MOV     [BP],AX                 ;SAVE NEW BLOCK COUNT
        DEC     AX                      ;DECREMENT BLOCK COUNT
        CALL    BDCONV                  ;BINARY TO DECIMAL CONVERSION
        MOV     DX,OFFSET BLKNUM        ;DX = BLOCK NUMBER MESSAGE
        MOV     AH,09H                  ;AH = 9 = DISPLAY BLOCK NUMBER
        INT     21H                     ;CALL DOS FOR SERVICE
        MOV     SI,OFFSET FILBUF        ;SI = FILE I/O BUFFER ADDRESS
        MOV     DI,OFFSET DPLINE+12     ;DI = HEXADECIMAL DISPLAY AREA
        MOV     BP,OFFSET DPLINE+62     ;BP = ASCII DISPLAY AREA
        MOV     CX,32                   ;CX = NUMBER OF 16 BYTE LINES
        PUSH    CX                      ;SAVE CX REGISTER ON THE STACK
        MOV     CX,16                   ;CONVERSION LOOP COUNTER
;
        PAGE
;
;       CONVERT A BYTE OF INPUT DATA TO HEXADECIMAL
;
DPF050:
        CMP     CX,8                    ;HALF OF LINE CONVERTED?
        JNE     DPF060                  ;BRANCH TO CONTINUE - NOT HALF
        DEC     DI                      ;DECREMENT LINE POSITION INDEX
        MOV     AL,'-'                  ;MOVE A DASH TO SEPARATE HALFS
        MOV     [DI],AL                 ;MOVE A DASH TO SEPARATE HALFS
        INC     DI                      ;INCREMENT LINE POSITION INDEX
;
;       SAVE CX ON THE STACK AND PROCESS NEXT BYTE
;
DPF060:
        PUSH    CX                      ;SAVE CONVERSION LOOP COUNTER
        MOV     CL,4                    ;CL = 4 = SHIFT COUNTER
        MOV     AL,[SI]                 ;AL = FILE BUFFER CHARACTER
        MOV     AH,AL                   ;AH = FILE BUFFER CHARACTER
        CMP     AH,20H                  ;COMPARE AH AN ASCII " "
        JL      DPF070                  ;BRANCH IF UNDISPLAYABLE
        CMP     AH,7EH                  ;COMPARE AH TO ASCII "~"
        JNG     DPF080                  ;BRANCH IF DISPLAYABLE
;
;       SET AH TO AN ASCII PERIOD // NON-DISPLAYABLE CHARACTER
;
DPF070:
        MOV     AH,"."                  ;AH = AN ASCII PERIOD
;
;       AH CONTAINS THE DISPLAYABLE ASCII CHARACTER
;
DPF080:
        MOV     DS:[BP],AH              ;MOVE CHARACTER TO ASCII AREA
        SUB     AH,AH                   ;AH = ZEROS // SET FOR CONVERT
        SHL     AX,CL                   ;POSITION UPPER CL BITS OF AL
        SHR     AL,CL                   ;POSITION LOWER CL BITS OF AL
        XCHG    AL,AH                   ;EXCHANGE CONTENTS OF AL AND AH
        PUSH    AX                      ;SAVE CONTENTS OF AX ON STACK
        MOV     BX,OFFSET ASCIIT        ;BX = ASCII TABLE ADDRESS
        XLAT    ASCIIT                  ;CONVERT CHARACTER TO HEX
        MOV     [DI],AL                 ;MOVE CHARACTER TO DISPLAY LINE
        INC     DI                      ;INCREMENT STORAGE INDEX
        POP     AX                      ;RESTORE ORIGINAL AX CONTENTS
        MOV     CL,8                    ;CL = 8 = SHIFT COUNTER
        SHR     AX,CL                   ;POSITION UPPER 8 BITS
        XLAT    ASCIIT                  ;CONVERT CHARACTER TO HEX
        MOV     [DI],AL                 ;MOVE CHARACTER TO DISPLAY LINE
        ADD     DI,2                    ;INCREMENT DI FOR NEXT BYTE
        INC     SI                      ;INCREMENT SI FOR NEXT BYTE
        INC     BP                      ;INCREMENT BP FOR NEXT BYTE
        POP     CX                      ;RESTORE LOOP CONTROL COUNTER
        LOOP    DPF050                  ;BRANCH TO CONVERT FULL LINE
;
        PAGE
;
;       CONVERT AND SETUP RELATIVE BUFFER OFFSET
;
        MOV     AX,HEXCNT               ;AX = CURRENT BUFFER OFFSET
        MOV     DI,OFFSET DPLINE+6      ;DI = DISPLAY LINE OFFSET
        PUSH    SI                      ;SAVE BUFFER INDEX ON STACK
        CALL    BDCONV                  ;BINARY TO DECIMAL CONVERSION
        POP     SI                      ;RESTORE BUFFER INDEX
        MOV     DI,OFFSET DPLINE        ;DI = DISPLAY LINE OFFSET
        MOV     AX,HEXCNT               ;AX = CURRENT BUFFER OFFSET
        MOV     CX,4                    ;CX = LOOP COUNTER (4 BYTES)
;
;       CONVERT HEXADECIMAL TO ASCII // BUFFER OFFSET
;
DPF090:
        PUSH    CX                      ;SAVE LOOP COUNTER ON STACK
        MOV     CL,4                    ;CL = 4 = SHIFT COUNTER
        PUSH    AX                      ;SAVE CURRENT BUFFER OFFSET
        AND     AX,0F000H               ;ISOLATE UPPER NIBBLE // 4 BITS
        XCHG    AL,AH                   ;EXCHANGE VALUES IN AL AND AH
        SHR     AL,CL                   ;POSITION 4-BIT NIBBLE
        XLAT    ASCIIT                  ;CONVERT TO HEXADECIMAL BYTE
        MOV     [DI],AL                 ;STORE CONVERTED BYTE // HEX
        INC     DI                      ;INCREMENT BYTE STORAGE INDEX
        POP     AX                      ;RESTORE CONTENTS TO AX
        SHL     AX,CL                   ;POSITION NEXT 4-BIT NIBBLE
        POP     CX                      ;RESTORE THE LOOP COUNTER
        LOOP    DPF090                  ;LOOP TO CONVERT ALL NIBBLES
```

```
;
;       DISPLAY THE LAST LINE CONVERTED
;
        ADD     HEXCNT,16               ;UPDATE HEX OFFSET COUNTER
        MOV     BX,OFFSET DPLINE        ;BX = HEXADECIMAL DISPLAY AREA
        MOV     CX,80                   ;CX = DISPLAY LINE BYTE COUNT
;
;       DISPLAY A CONVERTED BYTE OF DATA
;
DPF100: MOV     DL,[BX]                 ;DL = CONVERTED DATA BYTE
        INC     BX                      ;INCREMENT BX TO NEXT BYTE
        MOV     AH,02H                  ;AH = 2 = DISPLAY CHARACTER
        INT     21H                     ;CALL DOS FOR SERVICE
        LOOP    DPF100                  ;BRANCH TO DISPLAY NEXT BYTE
        POP     CX                      ;RESTORE BLOCK COUNTER
        LOOP    DPF110                  ;BRANCH TO PRINT NEXT LINE
        MOV     HEXCNT,0                ;RESET THE HEX OFFSET COUNTER
        MOV     DX,OFFSET FMFEED        ;DX = CR, LF, AND FF BUFFER
        MOV     AH,09H                  ;AH = 9 = WRITE DATA TO DISPLAY
        INT     21H                     ;CALL DOS FOR SERVICE
        MOV     BP,80H                  ;BP = 80H // TOP OF STACK
        JMP     DPF020                  ;BRANCH TO READ NEXT BLOCK
        PAGE
;
;       CONVERT AND DISPLAY NEXT 16 BYTES OF DATA
;
DPF110:
        PUSH    CX                      ;SAVE CONTENTS OF LINE COUNTER
        MOV     AX,2020H                ;AX = 2 ASCII SPACES
        MOV     CX,39                   ;CX = NUMBER OF WORDS TO SET
        MOV     DI,OFFSET DPLINE        ;DI = DISPLAY LINE OFFSET
;
;       MOVE ASCII SPACES TO THE DISPLAY LINE BUFFER
;
DPF120:
        MOV     [DI],AX                 ;MOVE SPACES TO DISPLAY LINE
        ADD     DI,2                    ;INCREMENT DI BY WORDS
        LOOP    DPF120                  ;BRANCH TO SET ALL WORDS
;
        MOV     DI,OFFSET DPLINE+12     ;DI = HEXADECIMAL DISPLAY AREA
        MOV     BP,OFFSET DPLINE+62     ;BP = ASCII DISPLAY AREA
        MOV     CX,16                   ;RESET LINE CONVERSION COUNTER
        JMP     DPF050                  ;BRANCH TO CONVERT NEXT LINE
;
DPFILE  ENDP
;
        PAGE
;
;       BINARY TO DECIMAL (ASCII) CONVERSION
;
BDCONV  PROC    NEAR
        MOV     CX,3                    ;CX = LOOP COUNTER
        SUB     SI,SI                   ;SI = ZERO // FOR INDEX
;
;       BINARY TO DECIMAL CONVERSION LOOP
;
BDC010:
        SUB     DX,DX                   ;DX = ZERO // FOR DIVISION
        DIV     BINDEC[SI]              ;DIVIDE FOR DECIMAL DIGIT
        ADD     AL,30H                  ;ADD 30H FOR ASCII NUMBER
        MOV     [DI],AL                 ;STORE ASCII NUMERIC DIGIT
        INC     DI                      ;INCREMENT FOR NEXT NUMBER
        ADD     SI,2                    ;INCREMENT SI FOR NEXT NUMBER

        MOV     AX,DX                   ;RESTORE REMAINDER TO AX
        LOOP    BDC010                  ;BRANCH TO CONVERT ALL DATA
;
        ADD     AL,30H                  ;ADD 30H FOR ASCII
        MOV     [DI],AL                 ;STORE ASCII NUMERIC DIGIT
        RET                             ;RETURN TO CALLER
;
BDCONV  ENDP
;
CODSEG  ENDS
        END     DPFILE
```

DUMP-2

```
File name = dpfcom.com

Block Number = 0000

0000  0000  E9 C8 02 46 69 6C 65 20-6E 61 6D 65 20 3D 20 24   ...File name = $
0010  0016  42 6C 6F 63 6B 20 4E 75-6D 62 65 72 20 3D 20 30   Block Number = 0
0020  0032  30 30 30 0D 0A 0D 0A 24-46 69 6C 65 20 6F 70 65   000....$File ope
0030  0048  6E 20 65 72 72 6F 72 0D-0A 24 46 69 6C 65 20 6E   n error..$File n
0040  0064  6F 74 20 66 6F 75 6E 64-0D 0A 24 0D 0A 24 30 31   ot found..$..$01
0050  0080  32 33 34 35 36 37 38 39-41 42 43 44 45 46 0D 0A   23456789ABCDEF..
0060  0096  0C 24 20 20 20 20 20 20-20 20 20 20 20 20 20 20   .$
0070  0112  20 20 20 20 20 20 20 20-20 20 20 20 20 20 20 20
0080  0128  20 20 20 20 20 20 20 20-20 20 20 20 20 20 20 20
0090  0144  20 20 20 20 20 20 20 20-20 20 20 20 20 20 20 20
00A0  0160  20 20 20 20 20 20 20 20-20 20 20 20 20 20 20 20
00B0  0176  20 20 20 20 20 20 20 20-20 20 20 20 20 20 20 20
00C0  0192  20 20 20 20 20 20 20 20-20 20 20 20 20 20 20 20
00D0  0208  20 20 20 20 20 20 20 20-20 20 20 20 20 20 20 20
00E0  0224  20 20 20 20 20 20 20 20-20 20 20 20 20 20 20 20
00F0  0240  20 20 20 20 20 20 20 20-20 20 20 20 20 20 20 20
0100  0256  20 20 20 20 20 20 20 20-20 20 20 20 20 20 20 20
0110  0272  20 20 20 20 20 20 20 20-20 20 20 20 20 20 20 20
0120  0288  20 20 20 20 20 20 20 20-20 20 20 20 20 20 20 20
0130  0304  20 20 20 20 20 20 20 20-20 20 20 20 20 20 20 20
0140  0320  20 20 20 20 20 20 20 20-20 20 20 20 20 20 20 20
0150  0336  20 20 20 20 20 20 20 20-20 20 20 20 20 20 20 20
0160  0352  20 20 20 20 20 20 20 20-20 20 20 20 20 20 20 20
0170  0368  20 20 20 20 20 20 20 20-20 20 20 20 20 20 20 20
0180  0384  20 20 20 20 20 20 20 20-20 20 20 20 20 20 20 20
0190  0400  20 20 20 20 20 20 20 20-20 20 20 20 20 20 20 20
01A0  0416  20 20 20 20 20 20 20 20-20 20 20 20 20 20 20 20
01B0  0432  20 20 20 20 20 20 20 20-20 20 20 20 20 20 20 20
01C0  0448  20 20 20 20 20 20 20 20-20 20 20 20 20 20 20 20
01D0  0464  20 20 20 20 20 20 20 20-20 20 20 20 20 20 20 20
01E0  0480  20 20 20 20 20 20 20 20-20 20 20 20 20 20 20 20
01F0  0496  20 20 20 20 20 20 20 20-20 20 20 20 20 20 20 20

Block Number = 0001

0000  0000  20 20 20 20 20 20 20 20-20 20 20 20 20 20 20 20
0010  0016  20 20 20 20 20 20 20 20-20 20 20 20 20 20 20 20
0020  0032  20 20 20 20 20 20 20 20-20 20 20 20 20 20 20 20
0030  0048  20 20 20 20 20 20 20 20-20 20 20 20 20 20 20 20
0040  0064  20 20 20 20 20 20 20 20-20 20 20 20 20 20 20 20
0050  0080  20 20 20 20 20 20 20 20-20 20 20 20 20 20 20 20
0060  0096  20 20 0F 00 20 20 20 20-20 20 20 20 20 20 20 20     ..
0070  0112  20 20 20 20 20 20 20 20-20 20 20 20 20 20 20 20
0080  0128  20 20 20 20 20 20 20 20-20 20 20 20 20 20 20 20
0090  0144  20 20 20 20 20 20 20 20-20 20 20 20 20 20 20 20
00A0  0160  20 20 20 20 20 20 20 20-20 20 20 20 20 20 20 20
00B0  0176  20 20 20 20 20 20 20 20-20 20 20 20 20 20 20 20
00C0  0192  20 00 0A 00 00 E8 03 64-00 0A 00 2B C0 BD 80 00    ......d...+....
00D0  0208  89 46 00 BC 00 01 BA 03-01 B4 09 CD 21 BA 62 03   .F.........!.b.

00E0  0224  B4 0A CD 21 BA 4B 01 B4-09 CD 21 BE 64 03 BF 5C   ...!.K....!.d..\
00F0  0240  00 2A C0 B4 29 CD 21 BA-5C 00 B4 0F CD 21 3C 00   .*..).!.\....!<.
0100  0256  74 09 BA 28 01 B4 09 CD-21 CD 20 BA 4B 01 B4 09   t..(....!. .K...
0110  0272  CD 21 BA 62 01 B4 1A CD-21 BF 5C 00 B8 00 02 89   .!.b....!.\.....
0120  0288  45 0E 2B C0 89 45 20 89-45 22 88 45 24 B9 00 01   E.+..E .E".E$...
0130  0304  BF 62 01 2B C0 89 05 83-C7 02 E2 F9 BA 5C 00 B9   .b.+.........\..
0140  0320  01 00 B4 27 CD 21 3C 00-74 07 83 F9 00 75 02 CD   ...'.!<.t....u..
0150  0336  20 BF 1F 01 8B 46 00 40-89 46 00 48 E8 C9 00 BA    ....F.@.F.H....
0160  0352  10 01 B4 09 CD 21 BE 62-01 BF 7F 03 BD B1 03 B9   .....!.b........
0170  0368  20 00 51 B9 10 00 83 F9-08 75 06 4F B0 2D 88 05    .Q......u.O.-..
0180  0384  47 51 B1 04 8A 04 8A E0-80 FC 20 7C 05 80 FC 7E   GQ........ |...~
0190  0400  7E 02 B4 2E 3E 88 66 00-2A E4 D3 E0 D2 E8 86 C4   ~...>.f.*.......
01A0  0416  50 BB 4E 01 D7 88 05 47-58 B1 08 D3 E8 D7 88 05   P.N....GX.......
01B0  0432  83 C7 02 46 45 59 E2 BE-A1 C3 03 BF 79 03 56 E8   ...FEY......y.V.
01C0  0448  66 00 5E BF 73 03 A1 C3-03 B9 04 00 51 B1 04 50   f.^.s.......Q..P
01D0  0464  25 00 F0 86 C4 D2 E8 D7-88 05 47 58 D3 E0 59 E2   %.........GX..Y.
01E0  0480  EB 83 06 C3 03 10 BB 73-03 B9 50 00 8A 17 43 B4   .......s..P...C.
01F0  0496  02 CD 21 E2 F7 59 E2 13-C7 06 C3 03 00 00 BA 5E   ..!..Y.........^

Block Number = 0002

0000  0000  01 B4 09 CD 21 BD 80 00-E9 22 FF 51 B8 20 20 B9   ....!....".Q.  .
0010  0016  27 00 BF 73 03 89 05 83-C7 02 E2 F9 BF 7F 03 BD   '..s............
0020  0032  B1 03 B9 10 00 E9 4E FF-B9 03 00 2B F6 2B D2 F7   ......N....+.+..
0030  0048  B4 C5 03 04 30 88 05 47-83 C6 02 8B C2 E2 EE 04   ....0..G........
0040  0064  30 88 05 C3 00 00 00 00-00 00 00 00 00 00 00 00   0...............
0050  0080  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0060  0096  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0070  0112  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
```

FORTRAN PITFALLS

The Joys and Frustrations of working with Version 1.0 of the IBM/Microsoft Compiler

LARRY PRESS,
AKI RUNCHAL, AND
CLEMENT TAM

REALITY

In spite of its initial promise, the PC hardware potential has still not been realized. To begin with, the 8087 socket remains empty. IBM is not supporting the 8087, and while several companies advertise 8087 products for the PC, they offer little or no software. The FORTRAN compiler would not use an 8087 even if one were plugged in.

The IBM PC's megabyte of memory is as unusable as the missing 8087. The 8088 CPU uses a segmented approach to memory addressing, with 64Kb segments. This in turn imposes three restrictions on the FORTRAN compiler. No labeled common block can be greater than 64K, the total of all data areas for a single program must also be less than 64K, and finally, any dimensioned array must be less than 32K. These limitations are restrictive, but they can be circumvented by splitting code into smaller modules. This has taken time, and in some cases it may not be feasible.

A more serious limitation for large codes is the fact that the present linker for the IBM-PC is unable to put together a program larger than approximately 300Kb. Even if a new linker were to eliminate this

The authors of this article are engineering consultants who provide services related to the development and application of technical software. Their work deals with fluid mechanics, heat and mass transfer, ground and surface water, air quality, and other process and environmental disciplines. Almost all of their software is written in FORTRAN, and until three years ago, they used large time-sharing networks with CDC, CRAY, and IBM mainframe systems. At that time, they acquired their first Z80-based CP/M system, and began using it for most of their software development, downloading the programs to the mainframe for final application runs. Soon they acquired two more Z80 based systems, one with a 10Mb Winchester.

Because they felt the IBM-PC offered significant advantages over their Z80-based systems in terms of the addressable memory, 16-bit registers, and processor speed, they ordered one as soon as it was announced. With the potential of the built-in 8087 socket and the promise of a Megabyte of memory, they thought they would be able to do almost all of their computing in-house. The following describes their experience, joys (a few), and frustrations (many) working with version 1.0 of the IBM-Microsoft FORTRAN compiler on the IBM PC. They write as users and software developers converting mainframe programs and developing new codes.

TABLE 1: WHETSTONE BENCHMARK STATISTICS

MACHINE	COMPILE TIME (sec)	LINK TIME (sec)	EXECUTION TIME (sec)	EXECUTABLE FILE SIZE (bytes)	REMARKS
Z80	28	72	128	14,000	CP/M, Microsoft FORTRAN version 3.34
IBM-PC	193	70	294	47,616	PC-DOS, IBM FORTRAN Default compiler option (RETURN statements removed)
IBM-PC	180	67	267	45,440	PC-DOS, IBM FORTRAN, $NODEBUG
IBM-PC	150	63	153	42,240	PC-DOS, IBM FORTRAN, $NODEBUG, $STORAGE:2
DEC			1.1		DECsystem 2060 (TOPS-20, F66)
DEC			2.1		VAX 11/780 (UNIX, F77)
PRIME			1.4		PRIME 750 (PRIMOS, F66)
PRIME			1.5		PRIME 750 (PRIMOS, F77)
APOLLO			6.2		10MHZ, MC68000, AEGIS 4.0, F77, floating point hardware
APOLLO			13.1		10MHz, MC68000, AEGIS 4.0, F77, software floating point
INTEL			16.0		8086/8087 Intel, FORTRAN
INTEL			390.0		8086, Intel FORTRAN

This table compares the memory requirements and execution speed of a standard benchmark for a number of different machines. An "old-fashioned" 4 mhz Z80 running the Microsoft FORTRAN compiler under CP/M outperforms the PC. We ran the Z80 and PC tests ourselves, but not the others; therefore, compile times and file sizes are not available for those cases.

restriction, the fact that IBM has reserved video buffer space starting at 640K might cause problems. Thus clever programming will be required to segment the program memory requirements to reach the 1Mb promise of the PC.

Another obstacle to compiling and running large programs is the capacity of the IBM disk drives. You really need a hard disk. Even with double-sided floppy drives, using the compiler is frustrating because the disks must be switched during the compile-link sequence. Among other things, a batch file cannot be used to control the compilation, tying you to the computer.

Efficiency

There are four dimensions to efficiency: compile and link time, memory required for compilation, object code size, and execution speed. After about 18 months experience converting and running a large number of programs, we cannot escape the conclusion that the FORTRAN compiler is inefficient in all four dimensions.

We can illustrate the performance of the compiler with three examples: a trivial one, a well established benchmark, and a large program we developed. For our tests, we compared the performance of the PC running FORTRAN under PC-DOS 1.1 with that of an "old-fashioned" Z80 machine running at 4MHz with Microsoft FORTRAN under CP/M.

The smallest possible FORTRAN program is merely this:

```
END
```

On the Z80 machine this program takes 59 seconds to compile and link and generates 1Kb code. On the IBM-PC it takes 120 seconds to compile and link, and the EXE file generated is 34,816 bytes long (with the debugging feature turned off, i.e., using the $NODEBUG option). This comparison illustrates the large base from which IBM-PC FORTRAN starts.

Our second test case was the Whetstone Benchmark, a standard benchmark developed in England in 1976. The comparison for the IBM-PC and Z80 machines is shown in table 1. The table also shows execution times for a number of popular minis. It is clear that the PC takes about two to three and one-half times longer to compile and link a program than the Z80, depending upon the compiler options selected. The code generated is about three times as large and executes about 20 percent slower than the comparable code on the Z80.

The third test was performed using one of our programs with about 2000 statements. Depending on the compiler options selected for this program, the compile and link time was from two and one-half to three and one-half times and the code size was approximately three times that generated on the Z80. The execution time was about 20 percent longer than on the Z80.

This large program, which consisted of 26 small subroutines, illustrated another dimension of the performance matrix. IBM FORTRAN is a two-pass compiler. The first pass generates a temporary code file, and the second generates optimized machine code. To our surprise we found that a small subroutine (50 statements) of this program required 256K of memory for compilation and then generated only 2Kb code. The memory was

If your thirst for technology is combined with an interest in the IBM Personal

ANNOUNCING

TECH PC JOURNAL

FOR IBM PERSONAL COMPUTER USERS

To The Limits... And Beyond

As you well know, the IBM Personal Computer is proving to be the success story of the 1980s. Only 18 months after its introduction, the IBM Personal Computer commands almost a fifth of all personal computer sales—a remarkable achievement. At present the number of IBM Personal Computers in users' hands exceeds a quarter-million, with perhaps as many as half a million on the way in the next 12 months.

But despite its widespread public acceptance and the wealth of products now available to support it, we are convinced that *vast areas of its potential remain unexplored,* not to mention those products waiting to be introduced for which the IBM Personal Computer is the precursor.

For those serious enough about microcomputers to accept the challenge, the IBM Personal Computer represents a host of untapped possibilities.

For those with a genuine appetite for technology, there are ways of stretching the IBM Personal Computer to the very limits of its specifications. Ways to willfully expand those specifications...to throw light into the dark recesses of its complex software systems. Ways to enlarge our understanding of the equipment and to *increase its usefulness.*

If this sounds appealing, then *PC Tech Journal* is for you. For it will be the definitive source of authoritative, technically-satisfying information about your IBM Personal Computer, its future transformations and the host of products developed to support it.

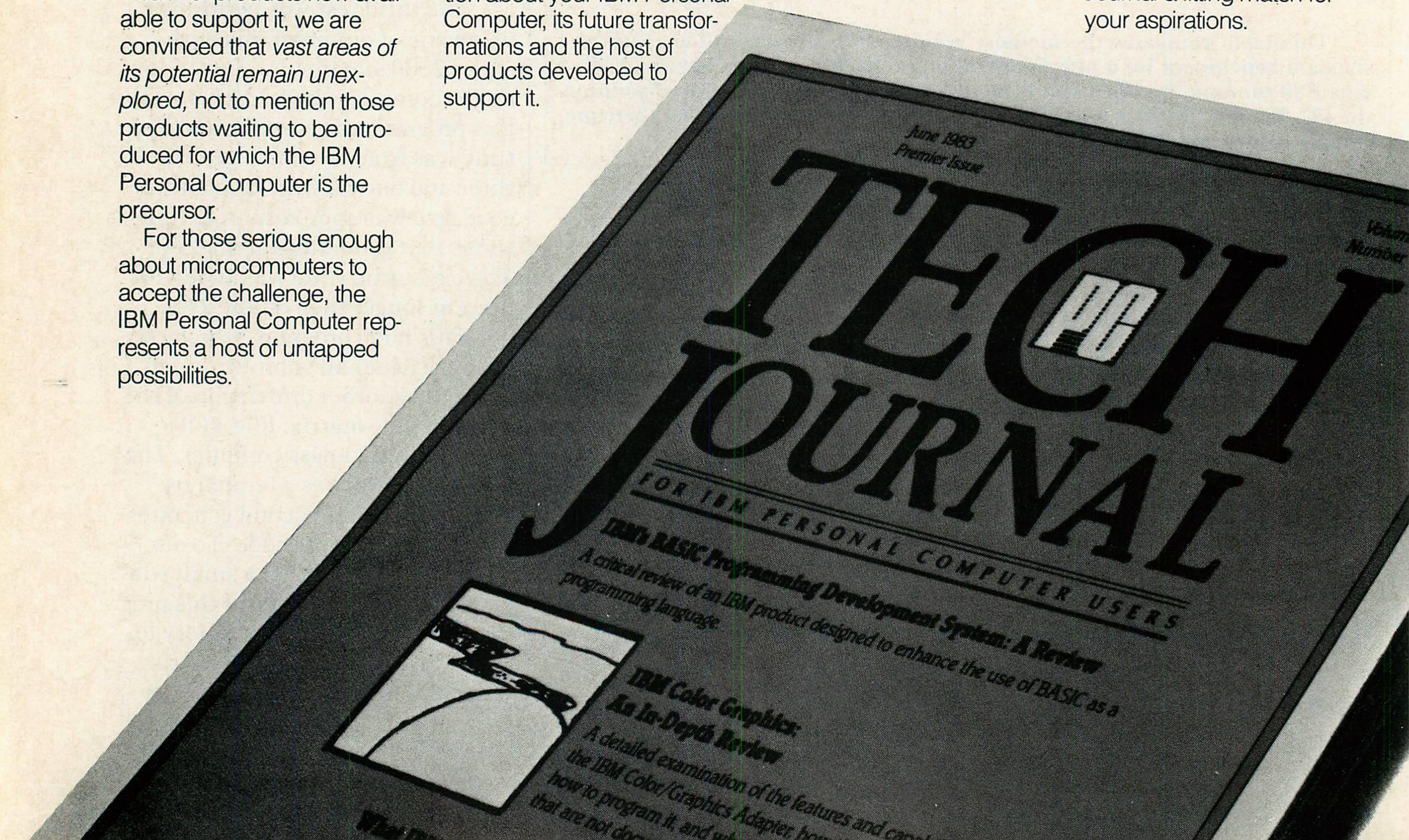
Why a Technical Journal?

The readership of *PC* Magazine has, in the last year, grown at a rate to match the success of the IBM Personal Computer itself. But as the number of committed readers has expanded, so has that portion of users who are not only interested in *output*—that is, using the IBM Personal Computer to better solve problems—but who care deeply about the *process* of the computer itself.

We consider you to be among that select group of readers. Your relationship with microcomputers, vocational or otherwise, has led you to levels of technical competence and curiosity which go far beyond problem-solving. In effect, you have a compelling interest not only in *what* your IBM Personal Computer does, but also *how* it does it and how it can do more.

For you, technological sophistication is a welcome challenge, perhaps even something of an adventure.

In both its depth and breadth of applicable sophistication, we are confident that you will find *PC Tech Journal* a fitting match for your aspirations.



used for extensive optimization tables, which is ironic given the slow execution times.

In general, we have found the compiler is anywhere from two to four times slower, and the executable code produced is two and one-half to five times larger than with Microsoft FORTRAN on a Z80 CP/M system. The compiler often required excessively large system memory and generated excessively large temporary files. One would expect that after all this two-pass optimization, the code generated would be efficient, and would run faster on the 4.77MHz 8088 with 16-bit arithmetic than on the old-fashioned 4MHz Z80 with 8-bit arithmetic, but, as the Whetstone Benchmark and our own experience indicate, that is not the case.

To make matters worse, the IBM-PC times quoted above are using a hard disk whereas those for the Z80 are for floppies. Thus the comparison would look even worse for the PC if one were to use floppy drives for the IBM (which, it bears repeating, is a most frustrating task).

It should be mentioned that changes in compiler options make significant difference in these statistics, but we have yet to find a case where IBM-PC was faster than the Z80. In fact the compiler options $DEBUG (turns on debugging mode) and $STORAGE:4 (increases word size from 2 to 4 bytes), are so demanding in memory and execution time that they are almost useless for any but small programs. There are other problems with the $DEBUG command as we will discuss under the BUGS section.

Shortcomings in the Implementation of Fortran-77

IBM-PC FORTRAN is a subset of the full FORTRAN-77 standard language. As such, it lacks some features useful for serious applications. The most important shortcomings are the absence of double precision and complex numbers. The latter can be circumvented by clever programming, but the former is a serious limitation. The installation of an 8087 along with a new compiler, would, of course, take care of this problem.

Table 2: Some inconvenient features of the Fortran-77 implementation

IMPLEMENTATION

1. Double Precision is missing.
2. Complex numbers are not allowed.
3. Use of Hollerith 'H' input in Data, Logical and Assignment statements is not allowed.
4. Character variables must be explicitly declared.
5. Character variables of different length cannot be compared.
6. One cannot equivalence different length character variables.
7. For unformatted write of array with length in each dimension declared, one must specify the ending index (or indices) in the implied do list.
8. Types cannot be mixed in declarations in COMMON.
9. One cannot initialize variables in COMMON block with DATA statements.
10. Formal parameters may not be initialized by DATA statements.
11. No more than 9 continuation statements can be used.
12. A comment statement cannot be embedded in a sequence of continuation statements.
13. Input records do not terminate upon receipt of CR-LF if the I/O list is not satisfied.
14. Multiple slashes do not generate extra line-feeds in output format control.
15. There is no "G" format.

PC FORTRAN is a subset of the full FORTRAN-77 standard. The subset implemented imposes certain restrictions which may prove inconvenient for those converting FORTRAN-66 programs to the PC. The absence of double precision and complex numbers are serious problems, and it was necessary to write a preprocessor to convert H format data to quoted strings. Most of the others are minor irritants; they don't send one's adrenalin level shooting up too fast.

There are other features of the subset that proved troublesome when we started converting our existing FORTRAN-66 codes to the IBM-PC. FORTRAN-77 is a more restricted language than FORTRAN-66, having added various type-checking constraints in keeping with trends in language design. This can cause problems for those of us who have grown up with FORTRAN-66 and gotten used to its forgiving syntax. Table 2 summarizes some of the complier limitations that caused problems during code conversion. Some of these are permitted in the full language but omitted from the IBM subset.

Many FORTRAN-77 compilers have options to "revert" to FORTRAN-66 conventions, in order to ease conversion problems. This compiler can do that to some extent, but we wish that it could do more. For instance, the prohibition on the use of the Hollerith (H) format for character data is the most annoying. While many FORTRAN-66 compilers allow quoted character strings, the H format is the FORTRAN-66 standard. Thus anyone trying to write a portable program would have used H format, and would have to go through a tedious editing effort to convert it to the PC. We finally decided to write a preprocessor in BASIC to substitute quoted strings for Hollerith fields, but would have dearly welcomed a "$HOLLERITH66" metacommand in the FORTRAN compiler.

Bugs and Peculiarities

One of the most irritating features of the compiler is its requirement that the I/0 list *must* contain the number of fields (characters) declared in the FORMAT statement. This presents a serious problem for those of us who prefer to type data on our keyboards rather than 80-column punched cards! The compiler does not accept a carriage-return/line feed as termination of an input record, but declares fatal error #1252. We ended up having to write another preprocessor to

blank-pad console input to 80-characters.

Another peculiarity of the compiler is the interpretation of the slash for output format control.

Normally, multiple slashes are used for extra line-feeds; however, in the IBM version any number of consecutive slashes are interpreted as a single slash. To get around this one, we have to use multiple carriage control characters at the beginning of a record.

There is an array-related peculiarity. If the length in each dimension of an array is declared explicitly, a WRITE performed on the array may have implicit DO loops. However, the same WRITE will not work if the array dimensions are passed as arguments of a subroutine. Consider the following example:

```
PROGRAM IBMPC
DIMENSION A(2)
DATA A, NI/1.,2., 2/
OPEN (4,FILE='TEMP';
     STATUS='NEW';
     FORM='BINARY')
WRITE (4) A
CALL SUB1(A, NI)
STOP
END
SUBROUTINE SUB1(A, NI)
DIMENSION A(NI)
WRITE(4) A
RETURN
END
```

The compiler accepts the WRITE statement in the main program but not in the subroutine. To compile this test case successfully, you must change the subroutine WRITE statement to:

```
WRITE (4) (A(I), I=1,NI)
```

We also found some inexplicable error messages. In converting one of our programs, we came across an error number that is not listed in the manual. The "heap full" message is also frustrating. In the default mode, the compiler only allows a certain maximum number of calls to subroutines (variously estimated to be between 780 and 800, depending on the memory available). IBM suggests two solutions. One is to eliminate all RETURN (yes!) statements from your subroutines; the other is to supress debugging with the $NODEBUG metacommand.

There are other strange happenings. On several occasions we came across a perfectly legitimate code that would not compile, but generated a nonsense error message. To our surprise we found that the same code would compile correctly if we just moved the "offending" statement one or two statements away from its original location. If that wasn't enough, we also discovered that some spurious "syntax errors" could be eliminated by just moving the statement to the right a few spaces.

Conversely, the compiler has some problems detecting true errors. We found the $DEBUG option to be highly temperamental. At times it missed on both the line and the subroutine in which an error occurred. After spending a great deal of time experimenting with this "feature," we finally decided to do without it.

Another quirk of the compiler (mentioned in the manual) is its inability to accept a comment card at the end of a module. It expects to find some more code when it encounters a comment card as the last statement.

There are several other peculiarities. Table 3 is a list of bugs that we and others have compiled. IBM recently started distributing a patch to the compiler through their dealers. We got a kick out of the fact that the patch disk contains a READ.ME file with a list of 13 "programming clarifications" which seems to be a new euphemism for "bugs-we-have-not-fixed." This list of thirteen is included in Table 3, and "circumventions" for each are suggested in the IBM patch. Unfortunately, the other bugs are not all corrected by the IBM patch. For instance, in some cases the blank-padded input bug mentioned above (error #1252) still occurs.

The linker (both Version 1.0 and 1.1) supplied also seems to have problems. We have one large program that requires 170Kb of memory to execute. At one stage we recompiled the program and then tried to link it, with 320Kb of system memory installed. The linker generated a message saying a VM.TMP file was being created on disk, because the linker needed even more main memory to complete its job. That seemed okay, but to our surprise the program would not run properly. After wasting two days trying different permutations and combinations, we added an extra 64Kb memory board, and to our surprise, the program executed successfully! The moral of the story is to beware if you see the VM.TMP message, because the linker may have lost its way between writing the temporary file on the disk and reading it back.

The linker also has some difficulty detecting errors. For example, we compiled two separate subroutines, one with the default wordsize (4 bytes) and the other with 2-byte integers. Both contained the same common block with integers in it. One would expect the linker to point out our error, noting the length discrepancy between the common blocks, but it did not. Of course the program would not execute properly.

Some Nice Things

The IBM manual is generally excellent. We found a few small problems; for instance the cosine function is listed as COSIN instead of COS, and the difference between the AMOD and AMOD4 functions is not clear, but these are nits. It is well organized and clearly written.

Our only serious complaint is the lack of clarification of compiler error messages. The manual merely repeats the compiler's cryptic description of the error, and provides no indication of how to interpret the location of the error from the addresses given by the compiler. The use of the $DEBUG option, when it works, does not improve the messages.

The FORTRAN compiler has a number of metacommands that are pleasant surprises. The $DO66,

$INCLUDE, and $STORAGE metacommands are welcome. The FORTRAN-77 subset provides for the block IF-THEN-ELSE structure, writing to internal files (equivalent of the ENCODE/DECODE of modified FORTRAN-66 compilers), several options for runtime file binding to a unit, extensive I/O file manipulations, PAUSE statements and versatile FORMAT statements. These features are comparable to those available on mainframe computers. If it worked properly, $DEBUG would also be a pleasant bonus.

CONCLUSIONS

The compiler is versatile and the documentation, on the whole, is impressive and informative. One can hope that the future releases will correct many of the shortcomings listed above. We can not escape the conclusion that there was inadequate testing of this compiler. Compilers should be tested in user environments where significant program development and conversion take place.

We would suggest that any serious FORTRAN user have a PC with at least 256Kb of memory and a hard disk. Even then, one should plan to spend a lot of time and effort if any conversion is involved—make an honest estimate of the conversion effort, multiply it by pi, and then for good measure multiply the result by e; those two being the natural irrational numbers that crop up in all kinds of places. Our experience seems to justify this formula. We also suggest that one should use batch files for all compiling and linking.

The foregoing has been a rather less than kind review of the IBM's FORTRAN compiler. The comments should, however, be put in some perspective. We have, with a lot of patience, cursing, and sweat, managed to convert much of our software to the IBM PC. Further, in most cases, we have managed to run large (300Kb) programs in-house, having made useful runs taking upwards of 10 hours. True, they would only have taken a minute on a CRAY-1, but that would have cost $200. On the PC it cost us only the electricity; so it has paid for itself.

We still prefer to use the Z80 machine for most of our development, since it is faster and not likely to give us "phantom" errors. Our typical mode of operation these days is to develop and debug software on the Z80, then download it to the PC for execution runs. If we had known how long it would be before we could get an efficient compiler capable of utilizing the PC's memory and the 8087, we might have waited, but as soon as that compiler is available, we'll doubtless get another PC. ■

TABLE 3: BUGS AND PECULIARITIES

1. The "Heap Full" error (1201) is generated by multiple subroutine calls.
2. There is no checking for allocation of more than 64K of data space, arrays of larger than 32K, and common blocks larger than 64K.
3. The DATA statement may not initialize CHARACTER variables properly.
4. The intrinsic function CHAR may not work properly in a WRITE statement.
5. A carriage control after a write to unit * is ignored if it follows a PAUSE statement.
6. The final CR/LF is not sent to the output device when a file is closed.
7. An internal error, 711, is generated by the use of a character array element as the "frame" parameter in an OPEN statement.
8. A function name, while in the scope of its own function, cannot be referenced in an expression.
9. If a file is created from a program and a READ is performed immediately, the results are not as expected.
10. Error #172 occurs if a multi-character variable is assigned to a one-character variable.
11. The arithmetic IF statement does not always work with real expressions generated by input through READ statements.
12. The EXTERNAL statement, if used, must be the last specification statement in a module.
13. Users are left on their own to discover a number of "undocumented" functions in the FORTRAN library. Determining whether the functions will work for an application is also left to the user.
14. The multiplication operator does not work properly under some rare circumstances.
15. The compiler generates the wrong code in cases where an invisible temporary variable is needed at runtime.
16. Error #1252 is generated if an input record terminates with CR/LF as long as the I/O list is not satisfied.
17. The multiple slash carriage control option does not work properly.
18. The compiler may find "phantom" errors in legitimate code.
19. Unexplained error messages may appear on the screen.
20. The error location pointed out by the $DEBUG Option may be incorrect.
21. The WRITE statement does not always function correctly with arrays.
22. The EOF function does not work with a formatted file.
23. The linker may not link the object code correctly under some circumstances.

The first 13 bugs are included in the material supplied on the IBM patch disk. Numbers 14 and 15 were among the ones that were brought to our notice by other users, and the patch may have cured them. The rest have not been cured by the patch. Number 23 is not a problem of the compiler but that of the linker.

Larry Press, Ph.D., of the Small Systems Group, is editing a series of books on the IBM PC, which will be published by John Wiley and Sons.

Aki Ranchal, Ph.D., is a principle consultant at Analytic and Computational Research and has 13 years experience as a consulting engineer and professor.

Clement Tam, M.S., is a project engineer at Analytic and Computational Research and has three years experience as a consulting engineer and scientific programmer.

TECH NOTEBOOK 5

Inducing Cold Reboots

Making the PC test its memory on command

ARTHUR A. GLECKLER

When the PC is first turned on, it tests its memory, processor, system board, expansion boards, and peripherals. If the Ctrl-Alt-Del key combination is pressed, these tests are repeated, with the exception of the memory test.

BIOS executes the memory test procedure only during a "cold" boot (when the PC has just been turned on). Whenever a reboot of any kind occurs, BIOS has to determine whether the computer is executing a cold boot or a warm boot; it does this by checking a flag it sets when the computer is first turned on or when Ctrl-Alt-Del is used. If this flag is modified to indicate that a cold boot is in progress, the computer will do a complete self-test, including a memory test. In this way, the PC can be fooled into thinking that it has just been turned on.

When Ctrl-Alt-Del is pressed, BIOS puts the number 1234h into location 0040:0072 to indicate that it is executing a warm boot. It then jumps to a rebooting routine (at F000:E05B), which finds 1234h in the flag at location 0040:0072, and jumps past the memory test to the rest of the bootstrap code.

When the PC is first turned on, the flag at 0040:0072 contains 0; once the Ctrl-Alt-Del combination has been pressed, this location is modified to contain the 1234h flag.

To induce a cold reboot under program control, something other than 1234h must be put into the flag location. To do this, enter DEBUG and type:

E 0040:0072 ⟨enter⟩
0 ⟨enter⟩

The flag location now contains 1200h. Enter a jump instruction to the reboot routine:

A 100 ⟨enter⟩
JMP F000:E05B ⟨enter⟩
⟨enter⟩

(If using the debugger under DOS 1.1, use the enter command E 100 to enter the bytes EA 5B EO OO FO.)

Execute the program by typing:

G = 100 ⟨enter⟩.

The computer will act as if it had just been turned on, testing its memory, peripherals, expansion boards, and processor.

The program shown in the listing will automate the cold reboot process under BASIC. Writing a batch file, perhaps called COLDBOOT.BAT, containing

BASIC COLDBOOT.BAS

allows the boot to be initiated with a DOS command.

The procedure outlined here is no substitute for a full diagnostic, such as the one included with IBM's *Guide To Operations* manual or the Advanced Diagnostic Package. However, if all that is needed is a quick check of the system memory, this program will serve the purpose.

In this way, the PC can be fooled into thinking that it has just been turned on.

```
1000 '  Cold reboot program - full reboot with memory test
1010 DEF SEG = &h40                  'define segment for flag at 0040:0072
1020 POKE &H72,0                     'poke value<>34h into flag location
1030 DEF SEG = &HF000                'define segment for machine language call
1040 COLDBOOT = &HE05B               'define machine language call location and
1050 CALL COLDBOOT                   '  execute call at f000:e05b
```

An Experiment in Software Marketing: FREEWARE

A novel approach to beating the high cost of bringing a product to market

RICHARD FOARD

I raised an eyebrow last summer when my friend handed me a diskette containing the PC-TALK program and said, "Here, it's a communications program." My friend, an upstanding fellow not known for purveying illicit software, noticed my discomfort and assured me, "It's okay—it's Freeware." A little mystified, I took the program for a spin later that evening and was greeted with the message shown in figure 1. Noting the California address, I mused with the East Coast smugness that this was obviously another well-intentioned but half-baked Utopian scheme born of whatever they put in the air out there on the West Coast. I expected little from the program, since its developers thought so little of it they gave it away. I had a lot to learn about PC-TALK and about Freeware.

First, the program's human engineering and functionality put IBM's Asynchronous Communications Package to shame. As I used the program to turn my PC into a terminal and dialed up the Source and other timesharing services, transmitted and received text files, and operated my Hayes Smartmodem with ease, it became obvious to me that I was using a mature, well designed piece of software. Second, the Freeware concept was anything but a half-baked idea. It was the product of a very thoughtful look at the software publishing business by a man experienced in both computing and publishing. Finally, it was not intended as a software giveaway program, but rather as an experimental, profit-making approach to the PC software business.

> *The program's human engineering and functionality put IBM's asynchronous communications package to shame.*

PC-TALK was born of Andrew Fluegelman's frustrated attempts to modify IBM's Asynchronous Communications package to meet special requirements. Fluegelman,

Richard Foard is vice president of software development of Travel Technology Corporation.

Figure 1:
Freeware's civil request for a "contribution"

If you have used this program and found it of value your contribution ($25 suggested) will be appreciated

—Freeware—
The Headlands Press, Inc.
P.O. Box 862
Tiburon, CA 94920

Regardless of whether you make a contribution, you are encouraged to copy and share this program.

Copyright © 1982 The Headlands Press, Inc.

founder of the Headlands Press and a veteran of ten years in publishing, was using his brand new IBM PC to collaborate with a co-author on a new book. His co-author was using a NorthStar machine for text preparation, and the two wished to exploit electronic communication by exchanging drafts over the telephone. Fluegelman spent one evening experimenting with modifications to the IBM package to resolve their interfacing problems and gave up. He embarked on the construction of his own general-purpose PC communication program the following day.

Because users had not been required to invest the purchase price before trying the product, the tone of complaints and suggestions was friendly and cooperative instead of demanding and abrasive.

Over the next few months, PC-TALK grew, evolving from a rudimentary program that met a special need to a formidable communications control program for the PC. Associates took note of the fruits of Fluegelman's efforts and began suggesting that he market the program. It was at this point that a number of ideas converged on him. His experience in the book publishing business left him with an appreciation for the headaches of bringing a published product to market by conventional means. Distribution channels would have to be selected and cultivated, royalties negotiated, pricing and marketing policies set, advertising placed, and a support capability established. All this would be expensive, difficult, and time-consuming. Then there was the matter of copy protection. Conventional software publishing wisdom dictated that distribution diskettes be copy protected to discourage piracy, but Fluegelman lacked the time necessary to develop an airtight copy protection scheme. As he watched a PBS fundraiser on television, the term "Freeware" popped into his head, some reflections on the software market and the nature of piracy jelled, and a novel software distribution scheme was born.

Fluegelman published notices on the Source and Compuserve announcing that PC-TALK was available *free* to anyone who sent him a diskette. Recipients were requested to make a *voluntary* $25 contribution if they used and liked the program. It was a bold idea. At the risk of giving away the fruits of months of work for nothing, he sidestepped the expense of advertising (there was no advertising), the difficulty of copy protection (copying was encouraged) and the problem of assuring potential customers that the software was worth the price (it was free for the trying).

The response was surprising and gratifying. Requests arrived in volume and, as the installed base of PC-TALK copies grew, contributions began coming in. Fluegelman estimates that 70 percent of those requesting PC-TALK copies directly have made the suggested monetary donation. Another pleasant surprise came when users began sending back their comments and suggestions for changes and extensions to the program. Because they had not been required to invest the purchase price before trying the product, the tone of complaints and suggestions was friendly and cooperative instead of demanding and abrasive. His file of complimentary letters grew quickly. Fluegelman was nothing short of thrilled at the magnitude and character of his response.

At least one fortune 500 company has adopted PC talk as standard equipment in outfitting its PC workstations and dutifully sends off a check each time it installs a new PC.

As PC-TALK's installed base increased in size, the concept of Freeware grew and prospered. Fluegelman continued the maintenance of the program as feedback (or "freeback," in the coinage of one of his friends) poured in, developing an extended-function third version called PC-TALK III and raising the suggested contribution amount to $35. A full-time employee was

hired to handle distribution of PC-TALK to new requestors and to notify old (contributing) customers when new versions became available or corrections to weaknesses were discovered. To date, Freeware has been in touch with an estimated 3,000 individual users in one way or another, and Fluegelman estimates that as many as ten or fifteen thousand copies of the program may be in use at this writing. At least one Fortune 500 company has adopted PC-TALK as standard equipment in outfitting its PC workstations and dutifully sends off a check to Headlands each time it installs a new PC.

Freeware's unconventional approach to software marketing has raised a few unexpected problems. Some companies, though impressed with the program and quite willing to pay for it, have found that the idea of a voluntary contribution is anathema to their accounting departments, and have asked Headlands to send them an invoice, *please.*

Freeware's success has inspired some imitators. As many as a dozen other programs are now being distributed under similar terms. Some confusion, in fact, has arisen from the use by others of the term "Freeware," which Headlands reserves as a trademark. While Fluegelman encourages others to make use of his distribution concept, he asks that they describe their activities using the generic term "User-Supported Software."

Freeware's philosophy of software distribution is based on the assumption that most people are willing to pay for quality in a software product if it's there for a reasonable price.

Fluegelman is cautiously enthusiastic when discussing plans for the future. He may undertake the publication of a catalog of user-supported software from all sources, and is eager to learn of other efforts to distribute commercial-grade software on a user-supported basis. The activities of his own enterprise, Freeware, may be expanded to serve as a coordinating clearing house for user-contributed extensions to PC-TALK and, perhaps, broadened to include the distribution and support of other software packages.

The authors of the current best-seller, *In Search of Excellence,* point out that companies that go to great lengths to watch over their employees to prevent them from malingering, gold-bricking, and making off with company goods are misplacing their efforts. Their policing is effective against 3 to 5 percent of their workers, and alienates and needlessly restricts all the others. The simple fact appears to be that most people *really want* to do a

good job; companies that embrace and nurture the good intentions of 97 percent of their employees are apt to do much better than those that don't. Freeware's philosophy of software distribution is a similar turnabout of conventional thinking, based on the assumption that most people are willing to pay for quality in a software product if it's there for a reasonable price.

So far, it appears that Freeware is right in its positive assumptions about personal computer users. By and large, they're an honorable lot who'll pay a reasonable price for quality, voluntarily, and who will cooperate with product developers on a very friendly basis to boot. That, at least, is Andrew Fluegelman's experience with those who've contacted Headlands directly for a copy of PC-TALK. Perhaps the ultimate test of the User-Supported software concept hinges upon the reactions of all the third- and fourth-party recipients of PC-TALK, those who copy the program from a friend or associate's copy. These people aren't exposed to whatever subtle psychological mechanisms come into play when one sends one's diskette (and good name) off in the mail and gets back a valuable software package for free. Third-party recipients are the cutting edge of the Freeware experiment in that they outnumber direct recipients by a significant (and impossible to measure) margin. If *they* respond to Freeware with a check, the experiment's nothing less than a resounding success.

> *Perhaps the ultimate test of the user-supported software concept hinges upon the reactions of all the third-and-fourth-party recipients of PC-TALK.*

PC-TALK was ranked eighth in a recent Softalk magazine software popularity survey, sharing the top ten with commercial heavyweights like MicroSoft's Multiplan. Freeware is a profit-making, growing enterprise that enjoys excellent relationships with loyal customers. It has inspired a trend in software distribution that may grow to considerable dimensions. Most interestingly, it has mounted a creative attack on the difficult and emotionally charged problem of software piracy and come out ahead by making some optimistic assumptions about human nature and the psychology of the marketplace. In Fluegelman's words, his policy of unrestricted, free distribution is "the best copy-protection scheme going." In the last analysis, the best software distribution channel of all just may turn out to be the complex network of "friendly pirates" that comprises all personal computer users. ■

CIRCLE NO. 133 ON READER SERVICE CARD

Freeware Review

PC-TALK: Communications Control

A look at one of the most popular programs for the PC

PC-TALK is a communications control program for the IBM Personal Computer. Used at its simplest level of function, PC-TALK is a terminal emulation program, using which a PC may be used as a "dumb" terminal to remote services like the Source, or for teletype-style communications with other terminals or computers. PC-TALK III, the most current version of the program, provides capabilities going beyond this basic level of function, including a telephone dialing directory, automatic transmission and receipt of text and binary files, error-checking communication protocols, and smart modem support.

PC-TALK scores excellent marks both in the level and variety of function it provides and in the critically important area of human factors engineering, or "user friendliness."

At first glance, it may seem that there's just not that much to do in a communications control program. To make a PC emulate a dumb terminal, for example, a program must receive and transmit characters via the PC's communication port, sending characters that the user types at the keyboard, and displaying those received from the communications line. The problem is only this simple, however, when viewed at a most superficial level. Even to allow simple, dumb terminal-style use, a communication program must allow its user to configure his terminal to suit the other communicating system's require-

ments by selecting the correct set of parameters from the very large set of possible combinations of baud rate, character stop bits, parity-checking arrangements, and transmission conventions and protocols. Presenting this complex set of possibilities to a user and allowing him to choose among them is a task that is easily done wrong.

After establishing basic parameters, the user faces still another task if he is using the telephone to call a remote system: conversing with the modem. The various "smart" modems on the market today provide an impressive array of functions, but require that the user learn the modem's command set to make use of it.

In fact, a PC user establishing telephone communication with another computer system must converse with as many as four levels of intelligence. A user must: (1) inform the program he is using on his PC of the communications "ground rules" (echo or don't echo locally typed characters, observe XON/XOFF flow control or not, etc.), (2) configure the communications device on his system to obey the rules imposed by the other side regarding baud rate, character width, and so on, (3) command his smart modem

> *PC-TALK makes it convenient at all times for the user to communicate directly with PC-TALK instead of with the remote computer system.*

to establish the desired connection by placing a telephone call, and (4) converse with the remote system in terms it expects when the connection is finally established. Often there are still other intermediate levels of intelligence to be dealt with, such as carriers like TELENET which must be told what sort of terminal the user is calling with and where he wishes to be connected.

Each level of intelligence requires that the user know, in some sense, its "language." PC-TALK's human engineering is excellent because it hides much of this multilingual complexity.

PC-TALK's human engineering advantage over other communications programs such as IBM's Asynchronous Communications Package lies in its continuous accessibility to the user. PC-TALK makes it convenient at all times for the user to communicate directly with PC-TALK instead of with the remote computer system; most commands require only a single, 'Alt'-shifted keystroke. Even in mid-conversation with another computer, for example, the PC-TALK user may request an on-screen display of a menu of PC-TALK commands. More important, the user may change communication parameters "on the fly," reconfiguring his local communications

device or changing ground rules without breaking connection with the other end of the conversation. Alone, this one simple capability gives PC-TALK a powerful advan-

> *With four or five keystrokes, a user who has a smartmodem can call the directory to the screen, select a telephone number, and dial it via a third-party long-distance service.*

tage over the IBM package, which requires that communication parameters be set, once and for all, before a user establishes a connection with another computer. If the IBM user guesses wrong, he must break the connection, reset the parameters, and try again. By contrast, a PC-TALK user who realizes he should have requested that his typed characters be echoed to his display when he made his call can, with one keystroke, correct his mistake and continue working with the remote system.

PC-TALK's dialing directory allows its user to call to the screen a previously stored catalog of 60 frequently called telephone numbers. With four or five keystrokes, a smartmodem-equipped user can call the directory to the screen, select a telephone number, and dial it (optionally via a third-party long distance service such as Sprint or MCI).

Each entry in the directory contains preset communications parameters as well as a telephone number, relieving the user of the tedium of remembering and re-setting the correct parameters each time he calls a number. The contents of the directory may be extended or changed using a straightforward set of directory maintenance commands. Like most PC-TALK functions, directory maintenance may be performed at any time, even in mid-conversation with another computer. The directory's ability to store and recall sets of communication parameters makes it useful even for users with modems lacking auto-dial capability.

PC-TALK makes the default assumption that is it dealing with a Hayes Smartmodem. Users may reconfigure the program to work with other modems, smart or dumb, by changing its modem commands. This ability exemplifies another of PC-TALK's strengths: it isn't presumptuous about the user's set of equipment or his working style. PC-TALK is configurable with respect to just about anything that

> *PC-TALK's excellent human engineering and functional power are most apparent in its file exchange capabilities.*

can vary from user to user.

PC-TALK's excellent human engineering and functional power are most apparent in its file exchange capabilities. A user may command PC-TALK to store incoming data in a disk file as well as displaying it on the screen by giving a "receive file" command and providing the name of the local disk file. Similarly, he may transmit a file stored on his local disk to a remote system by giving a single PC-TALK command. Both these capabilities may be used without knowledge of, or special treatment by, the remote system, which remains unaware that it is sending data to (or receiving data from) a disk file instead of a human being at a data terminal. File transmissions may be "paced"—transmitted with a user-settable delay at the end of each line.

Two conversing PC-TALK users have at their disposal an even more versatile file exchange system that allows the transmission of binary as well as text files, and, optionally, provides a data integrity-checking protocol (the XMODEM protocol) which may be used with either type of file. This protocol, in particular, allows data interchange between the PC and a wide variety of other computers.

PC-TALK's friendly design is evidenced in its conduct of file transmissions. In addition to error-checking protocol, a user may request that a file be transmitted with the "messages" option, which causes PC-TALK to inform both ends of the conversation when file transmission starts and stops; nothing is left to guesswork. If a transmitting user starts sending a file under the XMODEM protocol before the receiver is ready to receive it, the transmitting PC-TALK informs its user that it's waiting, delaying transmission until the receiver is ready, then starting automatically.

Version III of the program is suitable for both personal and professional use.

PC-TALK is an excellent, mature program for terminal emulation or inter-computer communication. Version III of the program is suitable for both personal and professional use. Its human engineering is very strong, and its dialing directory and file exchange capabilities distinguish it as a functionally complete package. Distributed by the Headlands Press under the novel Freeware™ scheme of software distribution, which Headlands pioneered, the program is accompanied by a clearly written, comprehensive user's manual on diskette. PC-TALK merits at least an evaluation by anyone using the IBM PC for data communications. ■

Freeware, *The Headlands Press, Inc., P.O. Box 862, Tiburon, CA 94920. $35, or free if a disk and a stamped, self-addressed mailer are provided.*

Freeware Review

Sure, but can you make money at it?

User-supported software is a great idea, especially for the end user. But what about the author of the program? Is it possible to make money from a product which is free for the asking?

Much depends on the circumstances. Because the product will not be advertised, word-of-mouth recommendations must get the product on its way. Obviously, the package must fill a pressing need so successfully that people are compelled to talk about it. Without such "advertising" the program is likely to languish.

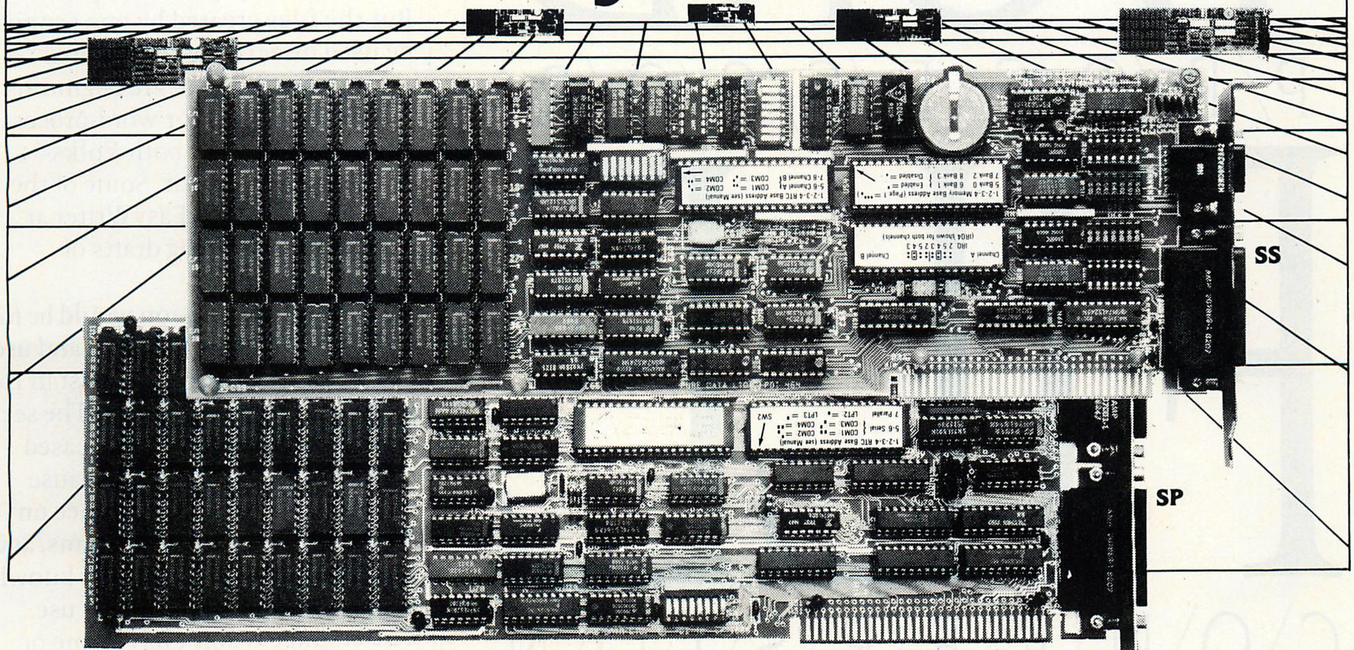
The support from the original source must be excellent. The author must make the acquisition process as painless as possible to encourage both direct sales and requests for free copies. (Knowing the number of free copies out can help in knowing the ratio of sold to free.) For paying customers, there must be support after the sale, especially for the repair of bugs.

All this means a considerable investment in time by the author, but the potential for reward seems to be there. Assume a modest goal of 1,000 copies of a program sold at a price of $30. The gross revenue is therefore $30,000. It is possible to produce a software package on disk and mail it for less than $6, so the basic cost of materials and shipping is $6,000. Spend another $1,000 to replace the diskette drives worn out by the copy process, $3,000 on general supplies, postage, and miscellaneous, and $20,000 is the gross profit before tax. That's a 33% overhead, or a margin of about 66%.

From a percentage standpoint, it's a winner. Any company showing a pre-tax profit of 66% is in great shape. But there is still one burning question: is it worth the author's time? A year of weekends and nights for a return of $20,000 might not be worth it. Then again, sales of 5,000 copieswould boost the return to $100,000, a not insignificant amount.

Can you make a buck this way? Given the right program at the right time, a quality product, word-of-mouth, and a reasonable distribution, the answer seems to be yes.—Will Fastie

Plug some Pure Data into your PC!



XT COMPATIBLE

PDI464-SP

- Up to 256K of RAM, fully socketed, with parity generation/checking
- Highest quality ceramic RAM chips used throughout
- IBM-compatible serial channel with selectable address and 6-foot cable
- IBM-compatible parallel printer adaptor with selectable address
- Real Time Clock/Calendar with field-replaceable lithium battery backup
- p-Disk™ disk emulator for single/double-sided diskette emulation
- Print spoolers for parallel and serial channels with multiple copies
- Software for clock support, dynamic memory configuration, memory testing
- Only occupies one expansion slot
- Fully-illustrated installation and operation manual

If you're looking for a multi-function board, you're faced with a bewildering array of functions, manufacturers, and prices. It's important to select the right card for your present and future needs. Pure Data offers a variety of high-quality expansion boards designed for the utmost in reliability and flexibility. We invite you to make comparisons with other brands. Our support software and documentation is the most complete available. Surprisingly enough, our products cost a lot less than you would expect. All of our IBM Personal Computer products are covered by a one year limited warranty, with a technical support hot-line and guaranteed 48-hour service.

PDI464-SS

- Up to 256K of RAM, fully socketed, with parity generation/checking
- Highest quality ceramic RAM chips used throughout
- Two IBM-compatible serial channels with selectable addresses, 6-foot cables
- Serial channels feature modem/printer configuration jumpers
- Real Time Clock/Calendar with field-replaceable lithium battery backup
- p-Disk™ disk emulator for single/double-sided diskette emulation
- Print spoolers for parallel and serial channels with multiple copies
- Software for clock support, dynamic memory configuration, memory testing
- Only occupies one expansion slot
- Fully-illustrated installation and operation manual

Pure Data Ltd

950 Denison Street, #17,
Markham, Ontario, Canada L3R 3K5
☎ (416) 498-1616 ☎ (416) 475-2424

Pure Data products are available through.

OnTrak DATA SYSTEMS
2974 LBJ Freeway, Suite 200,
Dallas, Texas 75234.
Telephone (214) 620-8000

Keysoft international, Computerland, Computer Innovations, The Answer and others.

WORD P/R/O/C/E/S/S/O/R FILE C\O\N\V\E\R\S\I\O\N

A BASIC program that provides a partial solution to the problem of converting text files among WordStar, VolksWriter, and EasyWriter.

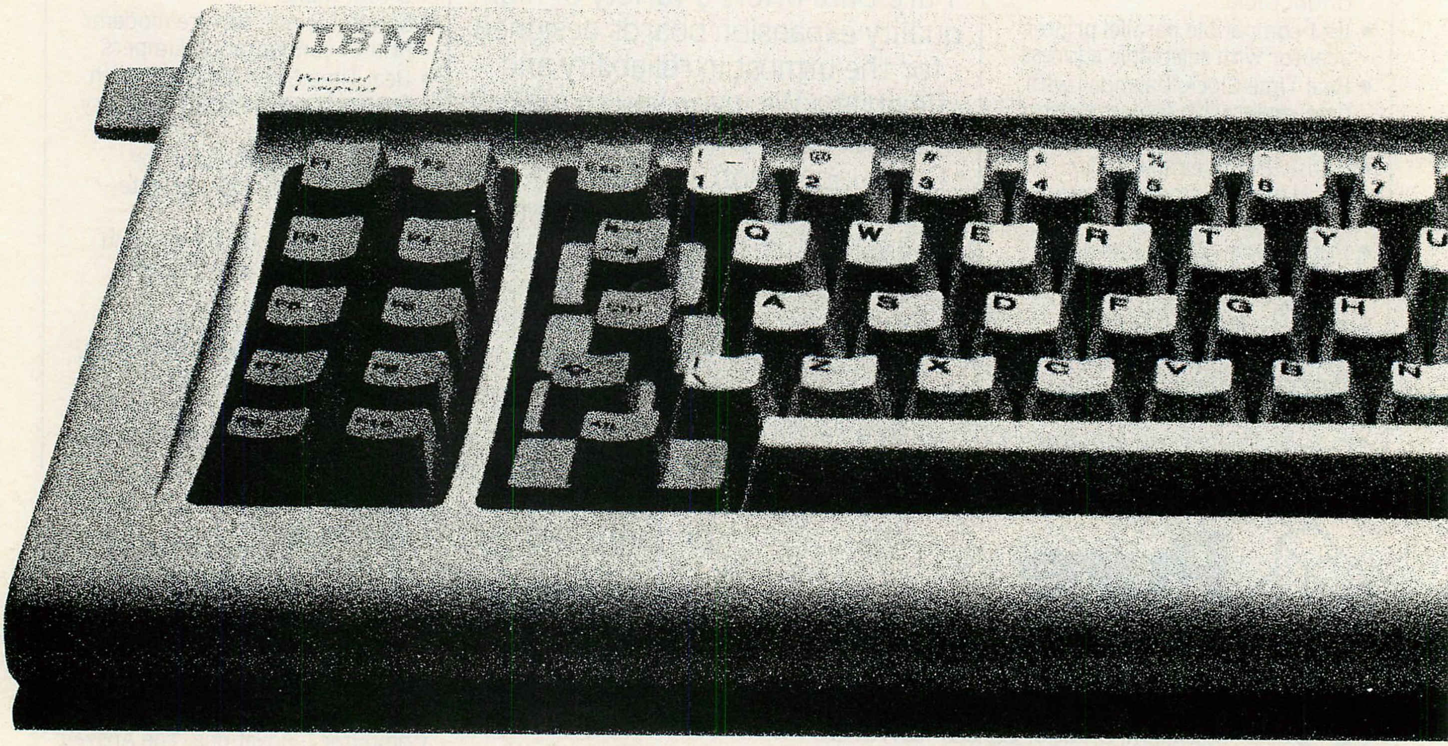
JIM GLASS

VolksWriter is simple to use and easy to learn. WordStar is powerful, but difficult to use and learn. EasyWriter II is popular and powerful. But the files created by one cannot be used directly by the others.

In my office, we are beginning to use an IBM PC for word-processing tasks. We have both VolksWriter and WordStar. Some of the staff members use EasyWriter at home for composing drafts of documents.

The ideal situation would be for the secretarial staff to learn and use WordStar and the technical staff to learn and use VolksWriter. The secretaries can handle the increased complexity of WordStar because they have previous experience on other word-processing systems, and because they will keep their knowledge fresh through constant use. The engineers don't have time or need to learn WordStar, and VolksWriter fits their need for a simple word-processor for draft documents.

The problem is that once an engineer has completed his or her



draft (which may be only a part of a larger document written by several persons), it cannot be read into WordStar. This is because VolksWriter, EasyWriter, WordStar, and other word processors each use different conventions for indicating paragraphs, ordinary lines of text, over-struck lines, and so forth. Actually, WordStar can read a VolksWriter file, but the text cannot be reformatted within WordStar, because each line is interpreted by WordStar to have a "hard" carriage return at its end. The VolksWriter end-of-paragraph marker is interpreted by WordStar as a ctrl-T (^T) symbol, so all occurrences of this symbol must be removed from the file after it is read in by WordStar.

Reading a WordStar document into VolksWriter is even worse: WordStar, for reasons known only to MicroPro, adds 128 to the ASCII values of some of the characters in the file. This apparently gives VolksWriter indigestion; it crashes when it tries to read an unmodified WordStar file.

Similar problems exist when reading a document EXPORTed from EasyWriter into either WordStar or VolksWriter.

Conversion of text files among WordStar, VolksWriter, and EasyWriter is a problem that anyone who owns more than one of these word-processors will face sooner or later. I have written a BASIC program to provide a partial solution to this problem. It is shown in listing 1, which follows this article. This program is intended to be compiled and run as an "EXE" file using the Basic Compiler. It will work under the interpreter, but be warned that the interpreted version is rather slow. I recommend the compiled version for production work, but you can eat lunch or read the paper if the need for conversion is infrequent.

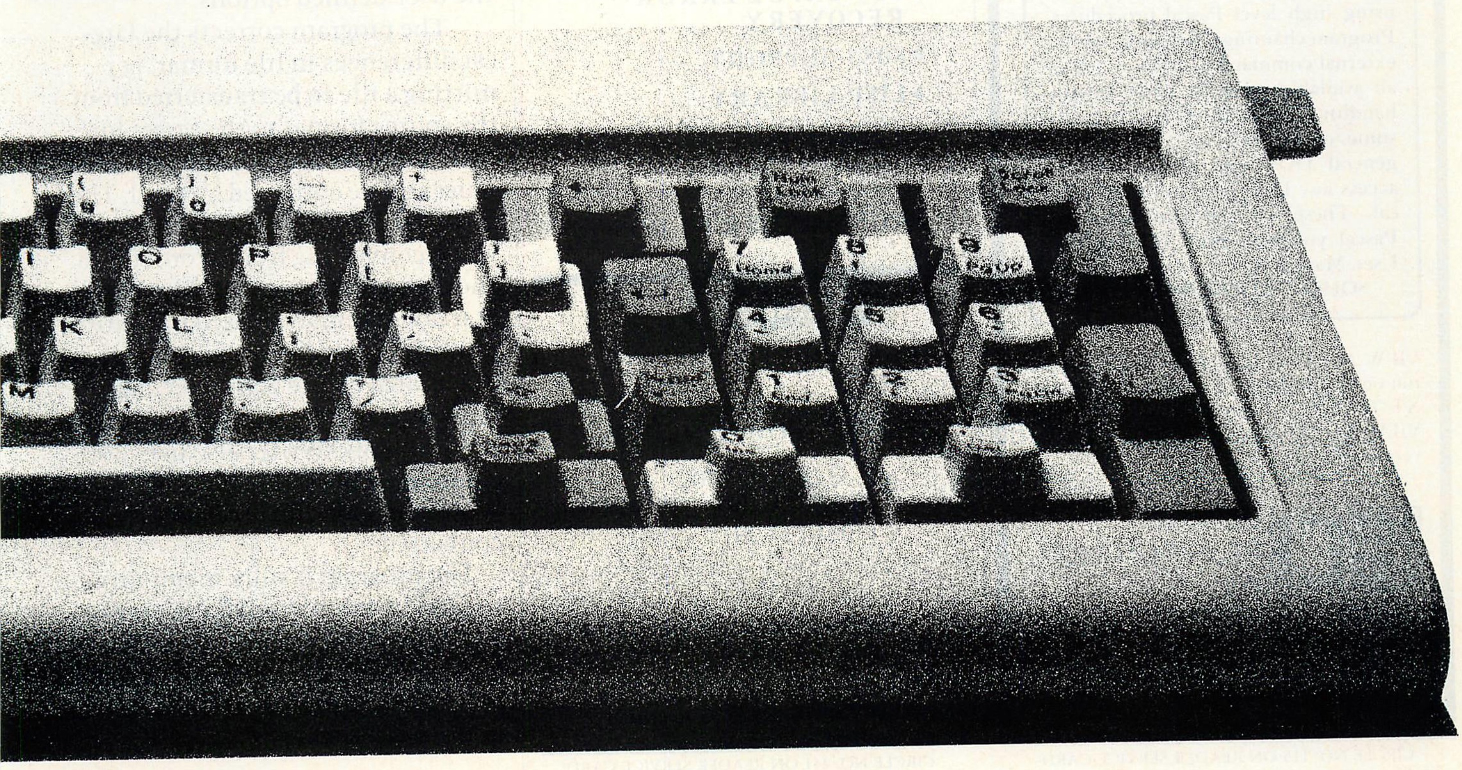
Reading a Wordstar document gives Volkswriter indigestion; it crashes when it tries to read an unmodified Wordstar file.

How it Works

The program works by reading the source file a character at a time. It builds a "test string" out of the incoming characters, and then tests it for sequences with special meaning to the "source" word-processor. (See table 1 for a summary of control sequences for the word-processors handled by this program.) It then substitutes characters that have equivalent functions under the "target" word-processor. Finally, the program builds a second string in memory by means of the BASIC "append" operator and dumps the string to the output file when its length approaches the maximum length permitted for a string variable.

The converter is table-driven; that is, all defined control character sequences are contained in tables that are searched for a match with

Jim Glass is a project engineer with the Rocketdyne Division of Rockwell Engineering, where he builds computer models of rocket engine and missile systems to optimize their design. He is also the president of his local computer club.



WORD FILE PROCESSOR CONVERSION CONTINUED

the incoming source test. The equivalent characters for the target word-processor are then substituted, again from tables. I have tried to minimize special-case coding in order to keep the converter as general as possible. However, there are a couple of instances where this was unavoidable. These cases included handling of WordStar's super-and subscript toggles (because the same character is used to both turn on and turn off the function) and discriminating a paragraph from a wrap-around line (because ASCII character 13, carriage return, occurs in both situations and sometimes "fooled" the converter).

These instances of "special coding" mean that you must be careful in using the "roll-your-own" options. If you read in a "user defined" table that is identical to, say, the VolksWriter table contained in the program, it will not function identically to the VW menu choice, due to the special code that depends on the index of the menu item. This is an unfortunate but unavoidable result of having to resort to special code, and somewhat limits the usefulness of the user-defined options.

The program corrects the largest differences in file format, permitting a file to be transferred from the source format to the target format. Some remaining differences must still be corrected by hand. The majority of such differences occur when converting from WordStar to another word processor. The WordStar dot commands, used to control printing of the document, cannot be handled by the converter, and must be fixed by the user after conversion. VolksWriter also uses dot commands, though not as many as WordStar, and these are also not converted.

The program is fully menu-driven, and I've tried to make it easy to use. It prompts the user for all inputs, and performs some error trapping. Among the features included in the program are as follows:


THE BUSINESS MANAGER™
A complete software solution
packaged on hard disk for the IBM® PC
GENERAL LEDGER
ACCOUNTS RECEIVABLE
BILLING
INVENTORY
ACCOUNTS PAYABLE
PAYROLL
WORD PROCESSOR - WORDSTAR™
SPELLING CHECKER - SPELLSTAR™
FILE MANAGEMENT - MAILMERGE™
ELECTRONIC SPREADSHEET- REPORT MANAGER™
Application generator plus 3-dimensional spreadsheet w/graphics
RELATIONAL DATABASE MANAGEMENT SYSTEM
MicroRIM™ (True mainframe capabilities)
★ All software installed on hard disk, initialized and ready to run.
★ Fully menu driven operating environment.
★ General Ledger comes with a chart of accounts and is fully interactive with Accounts Receivable, Billing, Inventory, Accounts Payable and Payroll.
★ Data may be directly transferred between MicroRIM, Report Manager and Wordstar.
★ Unlimited free phone consultation to dealers.
★ 90 days free phone consultation to end users.
★ 1 year warranty on all hardware.
IBM
Personal Computer Display
The Business Manager™
THE BUSINESS MANAGER™
1420 E. Edinger Ave., Suite 115, Santa Ana, CA 92705
(714)836-3560
ATTENTION DEALERS:
Call today to find out more about our dealer program
IBM is a Reg. trademark of IBM Corp. Wordstar, Spellstar and Mailmerge are Reg. trademarks of MicroPro International.
Report Manager is a Reg. trademark of Datamension Corp. Prices and specifications subject to change without notice.

WORD FILE PROCESSOR CONVERSION CONTINUED

- Support for four popular word processors, including WordStar, WolksWriter, EasyWriter, and EDLIN;
- Ability to accept a user-specified definition of up to two additional processors. Once specified, these definitions may be stored to disk and re-loaded at a later time.
- Estimates time required to process an input file based on length of the file. Also presents a running character/word count as processing proceeds. The time estimate is valid only if the program is run as a compiled rather than interpreted program.

The program does not convert every conceivable sequence of control characters from the source word-processor, but it does convert the most frequently used ones. The functions supported are

- Understriking
- Bold Face Text
- Shadow Text
- Superscripts
- Subscripts
- Word Wrap
- Paragraph Markers
- Strips WS "soft" hyphens, except at the end of a line

Other functions may be easily added to the program because of its table-driven nature.

EasyWriter and EDLIN

When using the program with EasyWriter II, conversion is limited by the way in which EasyWriter stores documents. Unlike other word processors, the IBM PC version of EasyWriter does not store its documents in standard DOS files. Instead, documents are stored in a large DOS file called a file folder, which may contain several documents at once. EasyWriter provides IMPORT and EXPORT functions, which can move a DOS file into a file folder or place a single document in a DOS file. However, these utilities work havoc on any format control characters in the documents.

On EXPORT, all EW control information is stripped from the document, leaving the resulting DOS file in plain vanilla EDLIN format. No end-of-paragraph, superscripting, boldfacing, etc., information is retained. On IMPORT, any characters that are not straight ASCII letters, numerals, or punctuation are replaced by a question mark ("?"), even those used by EasyWriter itself.

The converter thus does not attempt to do much to files specified as EW source files (i.e., those EXPORTED from EasyWriter), because there isn't any control information to convert. It will convert another source into an EW-compatible format suitable for IMPORT, but this will consist primarily of removing all those control sequences that EW would flag with a "?". If future releases of EW give the user the option of stripping or not stripping control characters, the program can easily be modified to take advantage of this.

EDLIN, while not really a word-processor, is also one of the programs supported. Because EDLIN does not use special control se-

Figure 1
Word-Processing Conversion Program—First Screen

```
Source—1 WORDSTAR (WS)
Target—2 VOLKSWRITER (VW)
       3 EASYWRITER (EW)
       4 EDLIN (ED)
       5 USER1 (US1)
       6 USER2 (US2)
Enter Index, Name, or abbreviation for source word-processor?
Enter Index, Name, or abbreviation for target word-processor?  2
Do you wish to see the text displayed as it is processed?   yes Strike any key to continue
```

Figure 2: File Selection and Start of Processing

```
Enter source ( WORDSTAR ) file name ? wsfil
Enter target ( VOLKSWRITER ) file name ? vwout1
wsfil is approximately 4480 characters in length. Estimated time to process this file is 1
MIN 39 SEC
Processing . . . 6 characters, 1 words. (Paragraph)
```

Figure 3 User-Specified Processor Prompts

```
Enter source ( VOLKSWRITER ) file name ? glass.art
Enter target ( USER1 ) file name ? glass.wp
glass.art is approximately 12735 characters in length. Estimated time to process this file
is 7 MIN 35 SEC
User-Defined Target Processor Definition
Do you wish to retrieve a definition ? n
Enter the number of characters used to denote the end of a wrap around line?  1 Enter
ascii (dec, hex, or octal) values of these 1 characters &015
Enter the number of characters used to denote the end of a paragraph?  2
Enter ascii (dec, hex, or octal) values of these 2 characters &HC, &HC
Enter the number of characters used to denote an overstruck line?  0
Enter the number of characters used to denote a boldfaced line?  1
Enter ascii (dec, hex, or octal) values of these 1 characters &0235
Enter the number of characters used to turn off a boldfaced line?  1
Enter ascii (dec, hex, or octal) values of these 1 characters &0234
Enter the number of characters used to denote an understruck phrase?  0
Enter the number of characters used to turn OFF understriking?
```

Table 1. Control Sequences

Word-Processor	Word-Wrap	Para-graph	Over-Strike	Bold-Face	Shadow Print	Under-Score	Super-Script	Sub-Script
WordStar								
DEC ASCII	141, 10	13, 10	13	02	04	19	20	22
Character	i, [LF]	[CR], [LF]	[CR]					
On Screen	[blank]	<	+	^B	^D	^S	^T	^V
Volkswriter								
DEC ASCII	13, 10	20, 13, 10	13, 13, 10	27, 69	27, 71	27, 45, 1	24	25
Character	[CR], [LF]	¶, [CR], [LF]	[CR], [CR], [LF]	[ESC], E	[ESC], G	[ESC], -		
On Screen	[blank]	¶	¶	[ESC]E	[ESC]G	[ESC]-		
EasyWriter								
DEC ASCII	(?)	(?)	(?)	(?)	(?)	(?)	(?)	(?)
Character	(?)	(?)	(?)	(?)	(?)	(?)	(?)	(?)
On Screen								
EDLIN								
DEC ASCII	13, 10	13, 10 13, 10	13	27, 69	27,71	27, 45, 1	27, 83, 00	27, 83, 01
Character	[CR], [LF]	[CR], [LF] [CR], [LF]	[CR]	[ESC],E	[ESC],G	[ESC]-	[ESC],S,	[ESC],S,
On Screen	[blank]	[blank]	[blank]	^E	^G	^-	^S	^S

NOTES:
"Character" refers to ASCII character as shown in Appendix G of the IBM BASIC Manual.
"On Screen" refers to appearance of control sequence on the screen of the source word-processor.
Question marks indicate unusual EasyWriter file structure. See text of article.
EDLIN (and some Volkswriter) character sequences assume EPSON printer.
Only **"ON"** sequences are shown for the sake of brevity. **"OFF"** sequences can be found in EPSON printer manual.
[CR] denotes carriage-return; **[LF]** denotes line feed.
[ESC] denotes ESCAPE (027): using actual escape will cause table to print incorrectly.

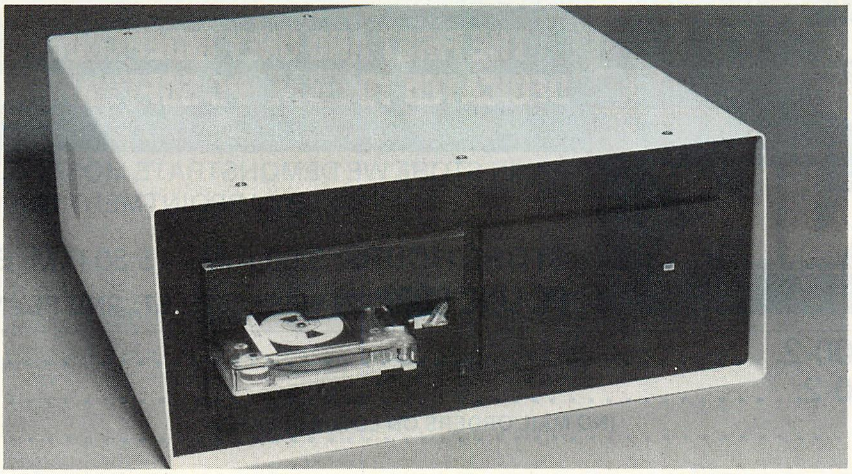
WORD FILE PROCESSOR CONVERSION CONTINUED

quences, the converter's internal tables are set up with standard EPSON control sequences. These can be read and written by EDLIN, and will cause the proper operations on Epson-type printers if the file is printed. Note that because EDLIN does not have a special way of indicating the end of a paragraph, the program assumes that paragraphs are denoted by two successive CR/LF pairs. This would be normal if paragraphs were separated by a blank line.

USER-INPUT DEFINITION

As I mentioned above, you may define up to two "roll your own" word-processors and thus customize the program for your own needs. When you specify either USER1 or USER2 (menu items 5 or 6) as one of the word-processors, the program will prompt you to enter the characters used by the word processor for the various functions. Character sequences are entered in two parts: first, the user enters the number of characters used for each function, then the ASCII values of these characters, separated by commas. The values may be entered in decimal, hex, or octal using the BASIC notation (&O for octal, &H for hex). You can also ask for a previously-stored table to be used, thus avoiding the tedious re-entry of this data. Tables read from the disk may also be modified to change just one control sequence. You can also write a new or modified table to disk.

When you complete the definition of your user-specified processor, the program will display a table, which shows the character sequences you've specified and their functions. In this display, the carriage-return (ASCII 13) is shown as a " symbol (ASCII 155), and the line feed (ASCII 10) is shown as an "L" symbol (ASCII 192). This substitution is made in order to make the normally-unprintable CR/LF pair printable.



If you think $1495 for a 10 Mbyte high quality hard disk is a good deal, then wait until your computer dealer tells you about our $695 25 Mbyte tape back-up.

The Pegasus 10 comes complete with all the software and hardware you need to start operations.

If you've outgrown the storage capabilities of your IBM PC or compatible computer but haven't grown into the giant pricetags on 10 Mbyte hard disk systems, you're in for a surprise.

The price on a complete 10 Mbyte system featuring the latest Shugart components just reached an all time low!

For the first time, just $1495 gets you everything you need to start working on your hard disk system:

- Top quality SA612 10 Mbyte (formatted) fixed hard disk (ST506 compatible)
- SASI controller card
- Software that runs on DOS 1.1 and 2.0, CP/M 86
- Host adapter
- Integral power supply
- Cables
- External custom chassis with additional space for either our tape backup, another hard disk, or floppy disk drive.
- One year warranty

What's the catch?

There is none. It's a simple matter of economics. Pegasus saw the growing need for mass storage, made a commitment to fill this need, purchased thousands of the finest quality hard disks, and is now passing the good deal along to you.

The only thing you may be missing in buying the Pegasus instead of the IBM XT expansion chassis is something you may not need in the first place: eight expansion slots, a communications card, three little initials, and an extra $2,000 out of your pocket.

What about support?

Pegasus and your dealer stand behind every Pegasus 10. If your unit requires maintenance anytime during the one year warranty, simply return it to your dealer for repair at no charge for parts or service.

But don't I need a tape backup?

We recommend it! That's why we're offering you the same great deal on a ¼" cartridge tape backup. We've watched other companies offer 10 Mbytes of storage and forget about the backup altogether. Our tape backup, which carries a 90 day warranty, retails for just $695 — a small price for over 25 Mbytes of formatted storage and lots of peace of mind.

Where can I buy a Pegasus 10 Mbyte system?

Pegasus systems are available only through dealers. So contact the dealer in your area who sells IBM PC or compatible computers. If he doesn't have Pegasus, chances are he'll carry it soon. Just ask him to call us. We will ship him your Pegasus 10 immediately.

Does Pegasus have larger storage systems?

Absolutely. 27, 65, and 140 Mbyte systems are soon available from Pegasus. And the best part is that they, too, are breaking new ground when it comes to pricing. If you've outgrown 10 Mbytes of storage, ask about the larger capacity units. All with the same top quality hardware and software and full one year warranty.

Shugart SA612 Hard Disk Specifications

12.95 Mbytes unformatted
10.416 Mbytes formatted
5.0 Mbits/set transferrable
16.2 msec access time track to track
92 msec access time average
345 tracks/inch
Mean Time Between Failure: 11,500 power on hours

Cipher 525-CT Tape Drive Specifications

32 Mbytes unformatted
28 Mbytes, 1024 byte sectors
6 recording tracks
Transfer rate (read/write) 500 Kbits/sec
78 ips streaming
¼ inch tape cartridge (3M DC 60A)
Standard floppy disk interface (SA450 or SA 850)

PEGASUS

2200 West Higgins Road • Suite 245
Hoffman Estates, IL 60195 • (312) 843-1090
Dealer inquiries invited.

CIRCLE NO. 122 ON READER SERVICE CARD

PC LINK™

THE FIRST AND ONLY INDEPENDENT STORE IN NYC DEDICATED TO IBM® PC SOFTWARE AND PERIPHERALS

IN STORE WE DEMONSTRATE MOST PRODUCTS WE SELL
BY APPOINTMENT ONLY

STORE HOURS: 9:00 A.M.—5:30 P.M. E.S.T MON.—FRI.
PC LINC CORP. 29 WEST 38TH ST., 2ND FL., NEW YORK, NY 10018

TANDON TM 100-2	$239.00
TANDON TM 50-2	CALL
LOTUS 1-2-3. (NO MAIL ORDERS ON LOTUS PLEASE)	CALL
TK SOLVER	CALL
QUAD LINK	CALL

BUSINESS SOFTWARE

APPLIED SOFTWARE	
VERSAFORM	$275.00
ASHTON-TATE	
dBASE II (PC DOS)	449.00
BOTTOM LINE STRATEGIST	319.00
FINANCIAL PLANNER	489.00
CONTINENTAL	
HOME ACCOUNTANT PLUS	95.00
FIRST CLASS MAIL	89.00
EAGLE SOFTWARE	
MONEY DECISIONS I & II	269.00
FOX & GELLER	
D. GRAPH	CALL
DIUTIL	CALL
QUICKCODE	CALL
HAYES	
SMARTCOM II	89.00
I U S	
EASY WRITER II	245.00
EASY SPELLER	139.00
EASY FILER	269.00
EASY PLANNER	189.00
LINK SYSTEMS	
DATAFAX	229.00
MICRO PRO	
CALCSTAR	CALL
INFOSTAR	CALL
WORDSTAR	CALL
WORDSTAR PROFESSIONAL	CALL
SPELLSTAR	CALL
MAILMERGE	CALL
MICROSOFT	
MULTIPLAN	199.00
PEACHTREE SOFTWARE	
PEACH GEN LEDGER	CALL
PEACH ACC RECV	CALL
PEACH ACC PAY	CALL
PEARLSOFT	
PERSONAL PEARL	190.00
SOFTWARE ARTS	
FINANCIAL MANAGEMENT	CALL
MECHANICAL ENGINEERING	CALL
SOFTWARE PUBLISHING	
P.F.S. REPORT	105.00
P.F.S. FILE	115.00
P.F.S. GRAPH	99.00
SORCIM	
SUPERCALC II	177.00
SOURCE	
SOURCE	75.00
SOUTHEASTERN	
DATA CAPTURE	115.00
T.M.Q.	
FILE FAX	145.00
TEXASOFT	
VERSATEXT	139.00

BUSINESS SOFTWARE Cont'd

VISICORP	
VISICALC	189.00
VISICALC ADVANCED	329.00
DESKTOP PLAN	219.00
VISIFILE	219.00
VISITREND/VISIPILOT	219.00

ENTERTAINMENT & EDUCATION

APPLE PANIC	$ 22.00
BUG OFF	29.00
COMPUTER FACTS AND FIVE	22.00
DEADLINE	39.00
FLIGHT SIMULATOR	35.00
FRIENDLY ARCADE	49.95
FRIENDLY PC INTRO SET	49.95
FROGGER	25.00
GORGON	29.00
HI RES #4	29.00
PC TUTOR	65.00
POOL 1.5	25.00
PROFESSIONAL BLACK JACK	60.00
SERPENTINE	25.00
SPACE STRIKE	25.00
STAR CROSS	29.00
SUSPENDED	39.00
ZORK I, II, III ea.	29.00
FUNTASTIC	
BIG TOP	29.00
COSMIC CRUSADERS	29.00
MASTER MINER	29.00
SNACK ATTACK II	29.00

(Call us for more games not listed here)

MONITORS

AMDEK 310A	$189.00
310G	179.00
COLOR I	CALL
COLOR II	CALL
NEC JC1203 RGB	759.00
QUADCHROME	CALL

DISKETTES

DYSAN	
S.S./D.D.	CALL
D.S./D.D.	CALL
KANGAROO (10 year warantee)	
S.S./D.D.	$ 22.95
D.S./D.D.	29.95
MAXELL	
S.S./S.D.	26.00
D.S./D.D.	44.00
VERBATIM (5 year warantee)	
21144 HEAD/CLEAN KIT	10.50
S.S./D.D.	26.00
D.S./D.D.	37.00

HARD DISK DRIVES

CORVUS HARD DISK 6, 11, 20 MB	CALL

A very special "Thank You to all of our patrons who helped in the making of P.C. Link's opening such a grand success!

EXPANSION BOARDS

USI COMPUTER MULTIDISPLAY CARD	$379.00
64 RAM CHIPS	59.00
AST CARDS	CALL
MONTE CARLO CARD	CALL
I-C-MAGIC CHIP	CALL
QUADBOARD CARD	CALL
AMDEK MAI CARD	CALL

MODEMS

HAYES SMART MODEM	$205.00
HAYES SMART MODEM 1200	485.00
HAYES SMART MODEM 1200B	CALL

ACCESSORIES

CURTIS PRODUCTS	
DISPLAY PEDESTAL	$ 65.00
MONO CABLE	45.00
KEYBOARD CABLE	35.00
DIABLO RIBBON	6.35
EPSON TWIN PAK RIBBONS 70/80	25.00
FLIP N FILE	24.50
KRAFT JOYSTICK	49.00
MBI JOYSTICK ADAPTER	12.00
DISKETTE CASES	2.95

PRINTERS

DIABLO P32 D-MATRIX	CALL
DIABLO P38 D-MATRIX	CALL
DIABLO 620	CALL
DIABLO 630	CALL
DIABLO 630 TRACTOR FEED	$249.00
C ITHO PROWRITER 10P	390.00
PROWRITER 10S	575.00
PROWRITER 15P	665.00
PROWRITER 15S	729.00
STARWRITER F-10	1250.00
TRACTOR FEED F-10	250.00
IDS PRISM 80	CALL
PRISM 132	CALL
OKIDATA 92P	649.00
93P	1009.00
NEC 8023	479.00
SANYO PR5500 DAISY PRINTER	655.00
PRINTER CABLE/PARALLEL	35.00

All prices listed in this ad are valid till Sept. 30, 1983. ALL BRANDS ARE REGISTERED TRADEMARKS. IBM IS A REGISTERED TRADEMARK OF IBM CORP.

CALL TOLL FREE 800-221-0343 CORPORATE ACCOUNTS WELCOMED **IN NEW YORK CALL 1-212-730-8036**

TERMS AND CONDITIONS

All prices reflect 3% cash discount. We reserve the right to repair, replace or return to manufacturer for repair, all goods acknowledged faulty or damaged on receipt by customer. *Customer must call for Return Authorization Number before returning any goods.* Prompt attention will be given to all damaged and faulty returned goods. Any goods returned for credit are subject to 10% restocking charge, plus shipping charge. No returns for credit on any software. Customer must deal with the manufacturer directly if the customer finds any false claims made by the manufacturer. Prices subject to change without notice. All advertised products are from our present stock and subject to availability. All goods are shipped U.P.S. surface unless otherwise authorized by customer. Add 2% of price, or minimum of $3.00 for shipping. We do not ship C.O.D. Please allow one to two weeks for personal or corporate checks to clear. To expidite shipping, send money order, certified or cashier's check, or charge to your **VISA—MasterCard.** (Minimum charge $50.00) N.Y.C. add 8¼% sales tax. N.Y.S. add appropriate sales tax. ***Terms, conditions and prices differ in our store.*** Prices subject to change without notice.

WORD FILE PROCESSOR CONVERSION CONTINUED

Program Operation

When you start the program, a menu is displayed as shown in Figure 1. You can select a source and a target word-processor from the list by index number, name, or by abbreviation. You cannot make both the source and target word-processors the same. If an index greater than six or less than zero is input, the program will re-prompt you for a valid input.

Figure 2 shows the next screen, which prompts for the names of the input (source) and output (target) files. At this point, the program prints a run-time estimate, and begins processing the input file. On color monitors, the word "Processing..." will flash in red at screen bottom, and the character and word counts will be displayed as shown.

Figure 3 illustrates the screen that comes up in response to a user-specified processor selection in the main menu. You may enter zero (0) for the number of characters if the function you are being prompted for is not present in your word processor. You may not enter zero for wrap-around and paragraph functions (unless you are modifying a definition read in from disk). This is because these functions must be defined for every word processor for the converter to work properly.

Conclusion

The conversion program presented here has solved my problem of inter-converting documents among the several that I and my co-workers use. It also represents an example of a class of table-driven text processing and modifying programs that can be used for other tasks in dealing with alphanumeric data. I hope that others will find it to be useful, if only as a study of what can be accomplished by a "brute force" BASIC program on the PC. ■

PCTECH JOURNAL PROGRAM LISTING #1

```
100 REM Word-Processor Interconversion Program
150 REM Copyright (C) 1983 by Jim Glass
200 REM
250 DEF FNUPS$(X$)=CHR$(ASC(X$)+32*(ASC(X$)>96 AND ASC(X$)<123))
300 REM
350 DEFINT A-Z
400 BLDFLG=0:SHDFLG=0:UNDFLG=0:SUPFLG=0:SUBFLG=0
450 X0=8:Y0=1
500 SH$=CHR$(31):SHI$=CHR$(30)
550 TRUE=-1:FALSE=0
600 N=6   'number of processors (up to 10) supported
650 DIM NWRP(10),NPAR(10),NOVR(10),NBLD(10),NUND(10),NSHD(10),NSUP(10),NSUB(10)

700 DIM NAM$(10,2),WRP$(10),PAR$(10),OVR$(10),BLD$(10),UND$(10),SHD$(10),SUP$(
    10),SUB$(10)
750 DIM A1(10),NBLDO(10),NUNDO(10),NSHDO(10),NSUPO(10),NSUBO(10)
800 DIM BLDOFF$(10),UNDOFF$(10),SHDOFF$(10),SUPOFF$(10),SUBOFF$(10)
850 GOSUB 25900 'initialize tables
900 LSTR=254
950 REM
1000 REM determine type of display
1050 REM
1100 DEF SEG=&HB800
1150 FOR I%=1 TO 255
1200 POKE &H3FFE,I%
1250 IF PEEK(&H3FFE)<>I% THEN 1500
1300 NEXT I%
1350 DEF SEG
1400 ICOLOR=1
1450 GOTO 1750
1500 ICOLOR=0
1550 DEF SEG
1600 REM
1650 REM get ready for first menu
1700 REM
1750 SCREEN 0,ICOLOR
1800 KEY OFF
1850 WIDTH 80
1900 IF ICOLOR THEN COLOR 6,0
1950 CLS
2000 GOSUB 2550 'MENU
2050 REM
2100 REM user processor was selected
2150 REM
2200 IF SOURCE>4 THEN PAS=1:GOSUB 5500 'user-defined processor is source
2250 IF TARGET>4 THEN PAS=2:GOSUB 5500 'user-defined processor is target
2300 REM
2350 GOSUB 11850 'file selection
2400 REM
2450 GOSUB 12600 'process the files
2500 END
2550 PRINT "Word-Processing Conversion Program"
2600 LOCATE 3,3:COLOR 3,0:PRINT "Word-Processors supported are:"
2650 FOR I=1 TO N:LOCATE I+4,12:PRINT I;NAM$(I,1);" (";NAM$(I,2);")";:NEXT

2700 LOCATE N+6,1:PRINT "Enter Index, Name, or abbreviation for";:COLOR 4,0:
     PRINT" source ";:COLOR 3,0:PRINT"word-processor";
```

```
2750 INPUT V$:FOR I=1 TO LEN(V$):MID$(V$,I)=FNUPS$(MID$(V$,I)):NEXT I
2800 SOURCE=INT(VAL(V$))
2850 REM
2900 REM if no number input, decode name
2950 REM
3000 IF SOURCE=0 THEN GOSUB 5150:SOURCE=INDX
3050 REM
3100 REM error-unrecognized name
3150 REM
3200 IF SOURCE<0 OR SOURCE>N THEN CLS:SOUND 200,2:GOTO 2550
3250 REM
3300 REM indicate source processor
3350 REM
3400 LOCATE SOURCE+4,3:COLOR 4,0:PRINT "Source-->";:PRINT SOURCE;NAM$(SOURCE,1)
     ;" (";NAM$(SOURCE,2);")";
3450 REM
3500 REM get target processor
3550 REM
3600 COLOR 3,0:LOCATE N+9,1:PRINT "Enter Index, Name, or abbreviation for";:
     COLOR 2,0:PRINT" target ";:COLOR 3,0:PRINT"word-processor";
3650 X1=CSRLIN:Y1=POS(0)+2
3700 INPUT V$:FOR I=1 TO LEN(V$):MID$(V$,I)=FNUPS$(MID$(V$,I)):NEXT I
3750 TARGET=INT(VAL(V$))
3800 REM
3850 REM if no number input then decode name input
3900 REM
3950 IF TARGET=0 THEN GOSUB 5150:TARGET=INDX
4000 REM
4050 REM check for invalid input
4100 REM
4150 IF TARGET>0 AND TARGET<N+1 THEN 4450
4200 SOUND 200,2:LOCATE X1,Y1:FOR I=1 TO LEN(V$):PRINT " ";:NEXT I
4250 GOTO 3600
4300 REM
4350 REM indicate target
4400 REM
4450 LOCATE TARGET+4,3:COLOR 2,0:PRINT "Target-->";:PRINT TARGET;NAM$(TARGET,1)
     ;" (";NAM$(TARGET,2);")";
4500 IF SOURCE=TARGET THEN SOUND 60,7:CLS:GOTO 2550
4550 LOCATE N+11,1:SOUND 1200,2:SOUND 1500,1:SOUND 800,2
4600 REM
4650 REM ask if display is desired
4700 PRINT "Do you wish to see the text displayed as it is processed ";
4750 INPUT A$:FOR I=1 TO LEN(A$):MID$(A$,I)=FNUPS$(MID$(A$,I)):NEXT I:IF A$=
     "Y" OR A$="YES" THEN FLG=-1 ELSE FLG=0
4800 REM
4850 REM edlin and ew require different paragraph strings on input than output
4900 REM
4950 IF TARGET=3 THEN NPAR(TARGET)=2:PAR$(TARGET)=CHR$(13)+CHR$(10)
5000 PRINT "Strike any key to continue"
5050 X$=INKEY$:IF LEN(X$)=0 THEN 5050
5100 RETURN
5150 REM convert name to index
5200 REM
5250 FOR I=1 TO N
5300 IF NAM$(I,1)=V$ OR NAM$(I,2)=V$ THEN INDX=I:RETURN
5350 NEXT I
5400 INDX=-999
5450 RETURN
5500 REM user-defined processor definition
5550 CLS:T$="User Input"
5600 MODFY=0
5650 IF PAS=1 THEN A$="Source":IDX=SOURCE ELSE A$="Target":IDX=TARGET
5700 PRINT "User-Defined ";A$;" Processor Definition"
5750 PRINT "Do you wish to retrieve a definition ";:INPUT A$
5800 FOR I=1 TO LEN(A$):MID$(A$,I)=FNUPS$(MID$(A$,I)):NEXT I
5850 RD=(A$="Y" OR A$="YES"):IF NOT RD THEN 7150
5900 PRINT"Enter file name for retreival of processor definition ";:INPUT T$
5950 OPEN T$ FOR INPUT AS #1
6000 INPUT#1,NWRP(IDX),NPAR(IDX),NOVR(IDX),NBLD(IDX),NUND(IDX),NSHD(IDX),NSUP(
     IDX),NSUB(IDX),NBLDO(IDX),NUNDO(IDX),NSHDO(IDX),NSUPO(IDX),NSUBO(IDX)
6050 'DUM$=INPUT$(2,#1) 'throw away cr/lf!
6100 IF NWRP(IDX)>0 THEN WRP$(IDX)=INPUT$(NWRP(IDX),#1)
6150 IF NPAR(IDX)>0 THEN PAR$(IDX)=INPUT$(NPAR(IDX),#1)
6200 IF NOVR(IDX)>0 THEN OVR$(IDX)=INPUT$(NOVR(IDX),#1)
6250 IF NBLD(IDX)>0 THEN BLD$(IDX)=INPUT$(NBLD(IDX),#1)
6300 IF NUND(IDX)>0 THEN UND$(IDX)=INPUT$(NUND(IDX),#1)
6350 IF NSHD(IDX)>0 THEN SHD$(IDX)=INPUT$(NSHD(IDX),#1)
6400 IF NSUP(IDX)>0 THEN SUP$(IDX)=INPUT$(NSUP(IDX),#1)
6450 IF NSUB(IDX)>0 THEN SUB$(IDX)=INPUT$(NSUB(IDX),#1)
6500 IF NBLDO(IDX)>0 THEN BLDOFF$(IDX)=INPUT$(NBLDO(IDX),#1)
6550 IF NUNDO(IDX)>0 THEN UNDOFF$(IDX)=INPUT$(NUNDO(IDX),#1)
6600 IF NSHDO(IDX)>0 THEN SHDOFF$(IDX)=INPUT$(NSHDO(IDX),#1)
6650 IF NSUPO(IDX)>0 THEN SUPOFF$(IDX)=INPUT$(NSUPO(IDX),#1)
6700 IF NSUBO(IDX)>0 THEN SUBOFF$(IDX)=INPUT$(NSUBO(IDX),#1)
6750 GOSUB 23450
6800 CLS
```

The Compare II Story

as told by
L.L. "Bill" Packer
Director of Engineering
Solution Technology, Inc.

About a year ago, our company had a problem we solved by writing a utility program called COMPARE. I was asked to tell the story about how COMPARE evolved and why you might want to have your own copy. This engineering story spans about one year.

Early in 1982, Solution Technology, received a contract to implement a hard disk file server and database system using IEE-488 communications links and Digital Research's MPM-II operating system. Here's the kicker, the project had to be done between the first of February and the middle of April. In all, I had five programmers working on seven different computer systems; some with hard disks and some with floppies. With so many machines, programs and archive floppies involved, plus an around - the - clock work schedule, all normal source control methods were a joke. Our two biggest problems were; what was broken when a program used to work, and who changed what when two programmers "fixed" something? I tried a number of different conventional control systems that only held the problem at bay. In the end, the control problem became so acute that I assigned a couple of my best software engineers to come up with a way to compare two source files. Well, they designed a traditional file compare program called, naturally, COMPARE, and that program made our software control problem at least manageable. COMPARE 1.1 was effective but slow, so we ran it mostly on the hard disks.

COMPARE 1.1 was a useful program, but it needed work. First we had to remove the file restrictions so differences between files could be ANY number of lines long. Second, we needed the program to be able to compare the two input files in something less than an eternity. So back into the lab we went to see what software technology could be applied. We added a pinch of database philosophy and a dash of communications technique, and the result was COMPARE 1.2 . . . a fast, line oriented file compare program which was very useful for programs and data files.

A short time later we were updating a technical manual for another contract and had to put change bars in each new issue to tell the end user where the changes were. Since text updates had been made by a number of people at different times, we attempted to use COMPARE 1.2 to locate all the places that had been changed. While somewhat successful, we still had to draw the change bars by hand. This was a huge waste of time because the process had to be repeated for every new issue. Enough! Time is money, so back into the lab. First to add word by word scanning and second, to be able to generate change bars automatically. Again success! We created COMPARE 1.3.

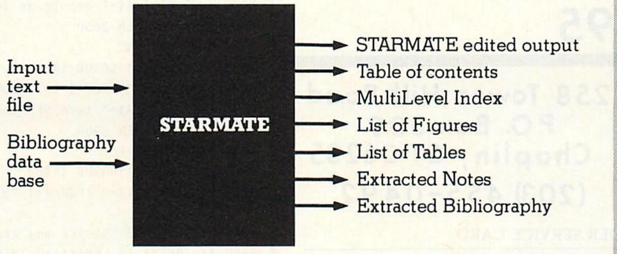
We have been selling COMPARE 1.3 for nine months now with hundreds of sales, excellent reviews, and customers telling us that we have a fine text analyzer. We've been pleasantly surprised at how many writers and lawyers have been using this package successfully. Even so, some said, "Why don't you have a version for the IBM-PC and CP/M-86?" We listened. During the first months of 1983, we returned to the lab and added the most asked for features to the CP/M-80 version then did the code conversions for CP/M-86 and PC-DOS. What resulted was COMPARE II. In COMPARE II we give the user the ability to customize for his own default file name extensions, internal editor file formats, strings for change bars, strings, for difference highlight on and off, printer width, output to console, full document side by side listing, definition of comments (to ignore) and much, much more.

Order COMPARE II directly from your local dealer or Solution Technology, 1499 W. Palmetto Park Road, Suite 218, Boca Raton, Florida 33432; (305) 368-6226. Single unit price is $145.00.

Get more than WordIndex for less $!

STARMATE

Users get high speed document finishing for WordStar with STARMATE under PC-DOS, CP/M-86, or CP/M 2.2.



Writers! Researchers! Engineers! Cut document makeup time from hours to minutes!

- Multi-Level Table of Contents
- Mulit-Level Index
- Lists of Tables and Figures
- Numbers Paragraphs (1.2, 1.3, etc)
- Prepares Footnotes
- Prepares Bibliography
- Clear commands, Numerous formatting options
- Reads documents with nested files

Specify When Ordering; Operating System, Computer Type and Disk Format. Free brochure and *nearly* free demo disk available.

STARMATE
(Special Introductory Price) **$145.00**
Demo All Disk (credits to purchase) $12.95

SOLUTION
TECHNOLOGY, INC.

"We **Deliver** Productivity"
1499 Palmetto Park Road
Suite 218
Boca Raton, FL 33432
305/368-6228

WordStar, PC-DOS and CP/M are trademarks of MicroPro, IBM and Digital Research respectively.

Check or COD, Florida residents add 5% sales tax. Dealer and Distributor inquiries welcome.

```
6850 PRINT:PRINT "Do you wish to modify the recovered definition ";:INPUT A$
6900 FOR I=1 TO LEN(A$):MID$(A$,I)=FNUPS$(MID$(A$,I)):NEXT I
6950 CLOSE 1
7000 IF A$<>"Y" AND A$<>"YES" THEN RETURN
7050 MODFY=-1
7100 PRINT"Enter zero for number of characters if no change"
7150 A$=" denote the end of a wrap around line":K=6:K1=3:GOSUB 23200
7200 IF NC=0 THEN 7400
7250 NWRP(IDX)=NC
7300 LOCATE 7,1:GOSUB 11200
7350 WRP$(IDX)="":FOR I=1 TO NWRP(IDX):WRP$(IDX)=WRP$(IDX)+CHR$(A1(I)):NEXT I
7400 K=9:K1=4:A$=" denote the end of a paragraph":GOSUB 23200
7450 IF NC=0 THEN 7650
7500 NPAR(IDX)=NC
7550 LOCATE 10,1:GOSUB 11200
7600 PAR$(IDX)="":FOR I=1 TO NPAR(IDX):PAR$(IDX)=PAR$(IDX)+CHR$(A1(I)):NEXT I
7650 K=12:K1=5:A$=" denote an overstruck line":GOSUB 23200
7700 IF NC=0 THEN 7900
7750 NOVR(IDX)=NC
7800 LOCATE 13,1:GOSUB 11200
7850 OVR$(IDX)="":FOR I=1 TO NOVR(IDX):OVR$(IDX)=OVR$(IDX)+CHR$(A1(I)):NEXT I
7900 K=15:K1=6:A$=" denote a boldface line ":GOSUB 23200
7950 IF NC=0 THEN 8150
8000 NBLD(IDX)=NC
8050 LOCATE 16,1:GOSUB 11200
8100 BLD$(IDX)="":FOR I=1 TO NBLD(IDX):BLD$(IDX)=BLD$(IDX)+CHR$(A1(I)):NEXT I
8150 K=17:K1=6:A$=" turn OFF a boldface line ":GOSUB 23200
8200 IF NC=0 THEN 8400
8250 NBLDO(IDX)=NC
8300 LOCATE 18,1:GOSUB 11200
8350 BLDOFF$(IDX)="":FOR I=1 TO NBLDO(IDX):BLDOFF$(IDX)=BLDOFF$(IDX)+CHR$(A1(I)
     ):NEXT I
8400 K=20:K1=7:A$=" denote an understruck phrase ":GOSUB 23200
8450 IF NC=0 THEN 8650
8500 NUND(IDX)=NC
8550 LOCATE 21,1:GOSUB 11200
8600 UND$(IDX)="":FOR I=1 TO NUND(IDX):UND$(IDX)=UND$(IDX)+CHR$(A1(I)):NEXT I
8650 K=22:K1=7:A$=" turn OFF understriking ":GOSUB 23200
8700 IF NC=0 THEN 8900
8750 NUNDO(IDX)=NC
8800 LOCATE 23,1:GOSUB 11200
8850 UNDOFF$(IDX)="":FOR I=1 TO NUNDO(IDX):UNDOFF$(IDX)=UNDOFF$(IDX)+CHR$(A1(I)
     ):NEXT I
8900 PRINT:PRINT "Strike any key to continue"
8950 X$=INKEY$:IF LEN(X$)=0 THEN 8950
9000 CLS
9050 K=4:K1=9:A$=" denote a shadowed phrase ":GOSUB 23200
9100 IF NC=0 THEN 9300
9150 NSHD(IDX)=NC
9200 LOCATE 5,1:GOSUB 11200
9250 SHD$(IDX)="":FOR I=1 TO NSHD(IDX):SHD$(IDX)=SHD$(IDX)+CHR$(A1(I)):NEXT I
9300 K=6:K1=9:A$=" turn OFF shadowing ":GOSUB 23200
9350 IF NC=0 THEN 9550
9400 NSHDO(IDX)=NC
9450 LOCATE 7,1:GOSUB 11200
9500 SHDOFF$="":FOR I=1 TO NSHDO(IDX):SHDOFF$(IDX)=SHDOFF$(IDX)+CHR$(A1(I)):
     NEXT I
9550 K=9:K1=11:A$=" denote a superscript ":GOSUB 23200
9600 IF NC=0 THEN 9800
9650 NSUP(IDX)=NC
9700 LOCATE 10,1:GOSUB 11200
9750 SUP$(IDX)="":FOR I=1 TO NSUP(IDX):SUP$(IDX)=SUP$(IDX)+CHR$(A1(I)):NEXT I
9800 K=11:K1=11:A$=" turn OFF superscripting ":GOSUB 23200
9850 IF NC=0 THEN 10050
9900 NSUPO(IDX)=NC
9950 LOCATE 12,1:GOSUB 11200
10000 SUPOFF$(IDX)="":FOR I=1 TO NSUPO(IDX):SUPOFF$(IDX)=SUPOFF$(IDX)+CHR$(A1(
      I)):NEXT I
10050 K=14:K1=3:A$=" denote a subscript ":GOSUB 23200
10100 IF NC=0 THEN 10300
10150 NSUB(IDX)=NC
10200 LOCATE 15,1:GOSUB 11200
10250 SUB$(IDX)="":FOR I=1 TO NSUB(IDX):SUB$(IDX)=SUB$(IDX)+CHR$(A1(I)):NEXT I
10300 K=16:K1=3:A$=" turn OFF subscripting ":GOSUB 23200
10350 IF NC=0 THEN 10550
10400 NSUBO(IDX)=NC
10450 LOCATE 17,1:GOSUB 11200
10500 SUBOFF$(IDX)="":FOR I=1 TO NSUBO(IDX):SUBOFF$(IDX)=SUBOFF$(IDX)+CHR$(A1(
      I)):NEXT I
10550 GOSUB 23450
10600 PRINT:PRINT "Do you wish to store this definition ";:INPUT A$
10650 FOR I=1 TO LEN(A$):MID$(A$,I)=FNUPS$(MID$(A$,I)):NEXT I
10700 IF A$<>"YES" AND A$<>"Y" THEN RETURN
10750 PRINT"Enter file name for storage of processor definition ";:INPUT A$
10800 OPEN A$ FOR OUTPUT AS #1
10850 PRINT#1,NWRP(IDX);NPAR(IDX);NOVR(IDX);NBLD(IDX);NUND(IDX);NSHD(IDX);NSUP(
      IDX);NSUB(IDX);NBLDO(IDX);NUNDO(IDX);NSHDO(IDX);NSUPO(IDX);NSUBO(IDX)
10900 PRINT #1,WRP$(IDX);PAR$(IDX);OVR$(IDX);BLD$(IDX);UND$(IDX);SHD$(IDX);SUP$
```

"No, we're all okay, but our computer's gone."

It could have been a lot worse.

Luckily, all that was missing was the computer and some spreadsheet software. And that should be covered by insurance.

But it isn't. Ed didn't realize that once he started using his home computer on business projects, his homeowners insurance no longer would cover it.

As suddenly as the robbery had occurred, Ed found himself out several thousand dollars with no way to recover it.

You need SAFEWARE™

If you use your microcomputer for business, you can protect yourself from a loss like this with SAFEWARE. For as little as $35 a year, SAFEWARE insurance covers all hardware, media and purchased software. Both business and personal.

SAFEWARE protects against theft, fire, accidental damage, even the destruction caused by power surges.

And, now, SAFEWARE is available on a 10-day trial basis. If you are not completely satisfied with the coverage provided, return the policy within 10 days for a full refund.

The only risk you take with SAFEWARE is not having it.

To obtain more information, or bind coverage immediately, call the toll free number. Or send the coupon to: Columbia National General Agency, 88 E. Broad St., Columbus, Ohio 43215 (In Ohio call toll free 1-800-848-2112).

1-800-848-3469

Hours: Mon.-Fri. 8:30-7; Sat.-Sun. 10:30-4:30 (EST)

SAFEWARE™

Amount of Insurance	Annual Premium
Up to $ 2,000	*$ 35*
$ 2,001-$ 5,000	*$ 60*
$ 5,001-$ 8,000	*$ 75*
$ 8,001-$11,000	*$ 90*
$11,001-$14,000	*$105*

Not avail. in AK, DC, HI, KY, LA, ME, MS, NV, SC, or WY.

Send for immediate protection: CNGA, 88 E. Broad St., Columbus, Ohio 43215

Name ____________________

Street ____________________

City ____________________ State ____________ ZIP ____________

System value $ ____________________

☐ Check enclosed ☐ VISA ☐ MasterCard Card # ____________

Expires: ____________ ☐ Send additional information

PCT

CIRCLE NO. 237 ON READER SERVICE CARD

```
(IDX);SUB$(IDX);BLDOFF$(IDX);UNDOFF$(IDX);SHDOFF$(IDX);SUPOFF$(IDX);
SUBOFF$(IDX);
10950 CLOSE 1
11000 RETURN
11050 REM
11100 REM parse input string to get values (up to 10)
11150 REM
11200 PRINT"Enter ascii (dec, hex, or octal) values of these ";NC;
      " characters ";
11250 FOR I=1 TO 10:A1(I)=0:NEXT:IF NC=0 THEN RETURN
11300 LINE INPUT V$
11350 K=INSTR(V$,","):IF K=0 OR NC=1 THEN K=LEN(V$)+1
11400 A1(1)=VAL(LEFT$(V$,K-1))
11450 IF NC=1 THEN RETURN
11500 FOR J=2 TO NC
11550 K1=INSTR(K+1,V$,","):IF K1=0 THEN K1=LEN(V$)
11600 A1(J)=VAL(MID$(V$,K+1,K1-K))
11650 K=K1
11700 NEXT J
11750 RETURN
11800 REM
11850 REM file selection
11900 REM
11950 CLS:COLOR 4,0:LOCATE 2,3:PRINT "Enter source ( ";NAM$(SOURCE,1);
      " ) file name ";
12000 INPUT F$
12050 OPEN F$ FOR INPUT AS 1
12100 LO#=LOF(1):LOCATE 6,3:COLOR 3,0:PRINT F$; " is approximately ";LO#;
      " characters in length."
12150 TSEC=.0185*LO#^1.0014:TMIN=TSEC\60:TSEC=TSEC-60*TMIN
12200 LOCATE 7,3:PRINT "Estimated time to process this file is " TMIN;" MIN ";
      TSEC; "SEC"
12250 LOCATE 4,3:COLOR 2,0:PRINT "Enter target ( ";NAM$(TARGET,1);
      " ) file name ";
12300 INPUT G$
12350 OPEN G$ FOR OUTPUT AS 2
12400 IF ICOLOR THEN COLOR 20,0 ELSE COLOR 31,0
12450 LOCATE 25,5:PRINT "Processing...";
12500 RETURN
12550 REM
12600 REM the main conversion routine
12650 REM
12700 L$=CHR$(10):C$=CHR$(13):X$="":Z$="":L=0:KK#=0:KW#=0
12750 ZZ$="" 'zz$ holds last character before present
12800 WHILE NOT(EOF(1))
12850 ON ERROR GOTO 16400
12900 Q$=INPUT$(1,#1) 'read source one character at a time
12950 ON ERROR GOTO 13000
13000 IF (Q$=" " OR Q$=C$) AND ZZ$<>Q$ THEN KW#=KW#+1 'count words
13050 REM
13100 REM fix wordstar funny business
13150 REM
13200 IF SOURCE=1 AND (ASC(Q$)>127 AND ASC(Q$)<>141) THEN Q$=CHR$(ASC(Q$)-128)
13250 REM
13300 Z$=Z$+Q$ 'build test string
13350 IF SOURCE=1 AND (Q$=SH$ OR Q$=SHI$) THEN SOFT=TRUE:MS=MS+1:LSH(MS)=LEN(Z$
      )
13400 REM
13450 REM scan test string for a match on each "special string"
13500 REM if found, branch to appropriate handler for this situation
13550 REM
13600 A=INSTR(Z$,WRP$(SOURCE)):IF A<>0 THEN GOSUB 17600:GOTO 14900
13650 B=INSTR(Z$,PAR$(SOURCE)):IF B<>0 THEN GOSUB 19350:GOTO 14900
13700 C=INSTR(Z$,OVR$(SOURCE)):IF C<>0 AND LEN(OVR$(SOURCE))<>0 THEN GOSUB
      19950:GOTO 14900
13750 REM
13800 REM load temp strings with control strings
13850 REM
13900 D$=BLD$(SOURCE):E$=SHD$(SOURCE):F1$=UND$(SOURCE):G1$=SUP$(SOURCE):H$=SUB$
      (SOURCE)
13950 REM
14000 REM see if flags are set for toggle
14050 REM
14100 GOSUB 22900
14150 REM
14200 REM look for controls in test string
14250 REM
14251 IF ASC(Z$)=21 AND SOURCE=2 AND TARGET=1 AND LEN(Z$)=1 THEN Z$=".pa":GOTO
      14900
14252 IF Z$=".pa" AND SOURCE=1 AND TARGET=2 THEN Z$=CHR$(21):GOTO 14900
14300 D=INSTR(Z$,D$):IF D<>0 AND LEN(D$)>0 THEN GOSUB 20400:GOTO 14900
14350 E=INSTR(Z$,E$):IF E<>0 AND LEN(E$)>0 THEN GOSUB 20800:GOTO 14900
14400 F=INSTR(Z$,F1$):IF F<>0 AND LEN(F1$)>0 THEN GOSUB 21300:GOTO 14900
14450 G=INSTR(Z$,G1$):IF G<>0 AND LEN(G1$)>0 THEN GOSUB 21800:GOTO 14900
14500 H=INSTR(Z$,H$):IF H<>0 AND LEN(H$)>0 THEN GOSUB 22350:GOTO 14900
14550 REM
14600 REM no controls found; go get another character
14650 REM
```

PC TECH JOURNAL

FREE INFORMATION SERVICE

Learning more about a product that's advertised or mentioned in an article in this month's issue is as simple as 1-2-3. And absolutely free.

1 Print or type your name and address on the attached card. Use only one card per person.

2 Circle the numbers on the card that correspond to the numbers at the bottom of the advertisements or articles for which you want more information. (Key numbers for advertised products also appear in the Advertisers' Index.)

3 Simply mail the card, and the literature will be mailed to you free of charge by the manufacturer.

The address on the attached card is for product information only. Editorial inquiries should be directed to *PC: Tech Journal*, One Park Avenue, New York, New York 10016.

PC TECH JOURNAL

Void after December 31, 1983

101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125
126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150
151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175
176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200
201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225
226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250
251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275
276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300
301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325
326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350
351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375
376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400
401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425
426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447 448 449 450
451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475
476 477 478 479 480 481 482 483 484 485 486 487 488 489 490 491 492 493 494 495 496 497 498 499 500

PLEASE PRINT CLEARLY—Use only one card per person.

NAME________________ PHONE # (____)________

COMPANY________________

ADDRESS________________ APT.______

CITY____________ STATE______ ZIP______

(Zip code must be included to insure delivery.)

4 ☐ **Please send me 1 year (6 issues) of *PC: Tech Journal* for $19.97 and bill me. (Full subscription price $24.97.)**

PCTJ9833

PC TECH JOURNAL

Void after December 31, 1983

101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125
126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150
151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175
176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200
201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225
226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250
251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275
276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300
301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325
326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350
351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375
376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400
401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425
426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447 448 449 450
451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475
476 477 478 479 480 481 482 483 484 485 486 487 488 489 490 491 492 493 494 495 496 497 498 499 500

PLEASE PRINT CLEARLY—Use only one card per person.

NAME________________ PHONE # (____)________

COMPANY________________

ADDRESS________________ APT.______

CITY____________ STATE______ ZIP______

(Zip code must be included to insure delivery.)

4 ☐ **Please send me 1 year (6 issues) of *PC: Tech Journal* for $19.97 and bill me. (Full subscription price $24.97.)**

PCTJ9832

PC TECH JOURNAL

Void after December 31, 1983

101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125
126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150
151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175
176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200
201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225
226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250
251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275
276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300
301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325
326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350
351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375
376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400
401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425
426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447 448 449 450
451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475
476 477 478 479 480 481 482 483 484 485 486 487 488 489 490 491 492 493 494 495 496 497 498 499 500

PLEASE PRINT CLEARLY—Use only one card per person.

NAME________________ PHONE # (____)________

COMPANY________________

ADDRESS________________ APT.______

CITY____________ STATE______ ZIP______

(Zip code must be included to insure delivery.)

4 ☐ **Please send me 1 year (6 issues) of *PC: Tech Journal* for $19.97 and bill me. (Full subscription price $24.97.)**

PCTJ9831

PUT STAMP HERE
THE POST OFFICE
WILL NOT DELIVER
MAIL WITHOUT
POSTAGE

TECH PC JOURNAL
P.O. Box 8590
Philadelphia, PA 19101

PUT STAMP HERE
THE POST OFFICE
WILL NOT DELIVER
MAIL WITHOUT
POSTAGE

TECH PC JOURNAL
P.O. Box 8590
Philadelphia, PA 19101

PUT STAMP HERE
THE POST OFFICE
WILL NOT DELIVER
MAIL WITHOUT
POSTAGE

TECH PC JOURNAL
P.O. Box 8590
Philadelphia, PA 19101

FREE

FREE

FREE

FREE

FREE

FREE

FREE

FREE

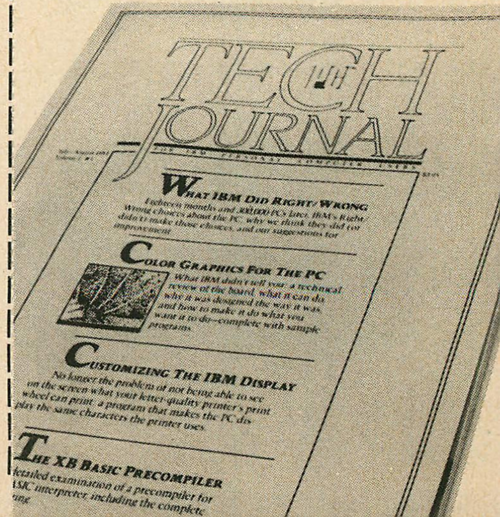
FREE

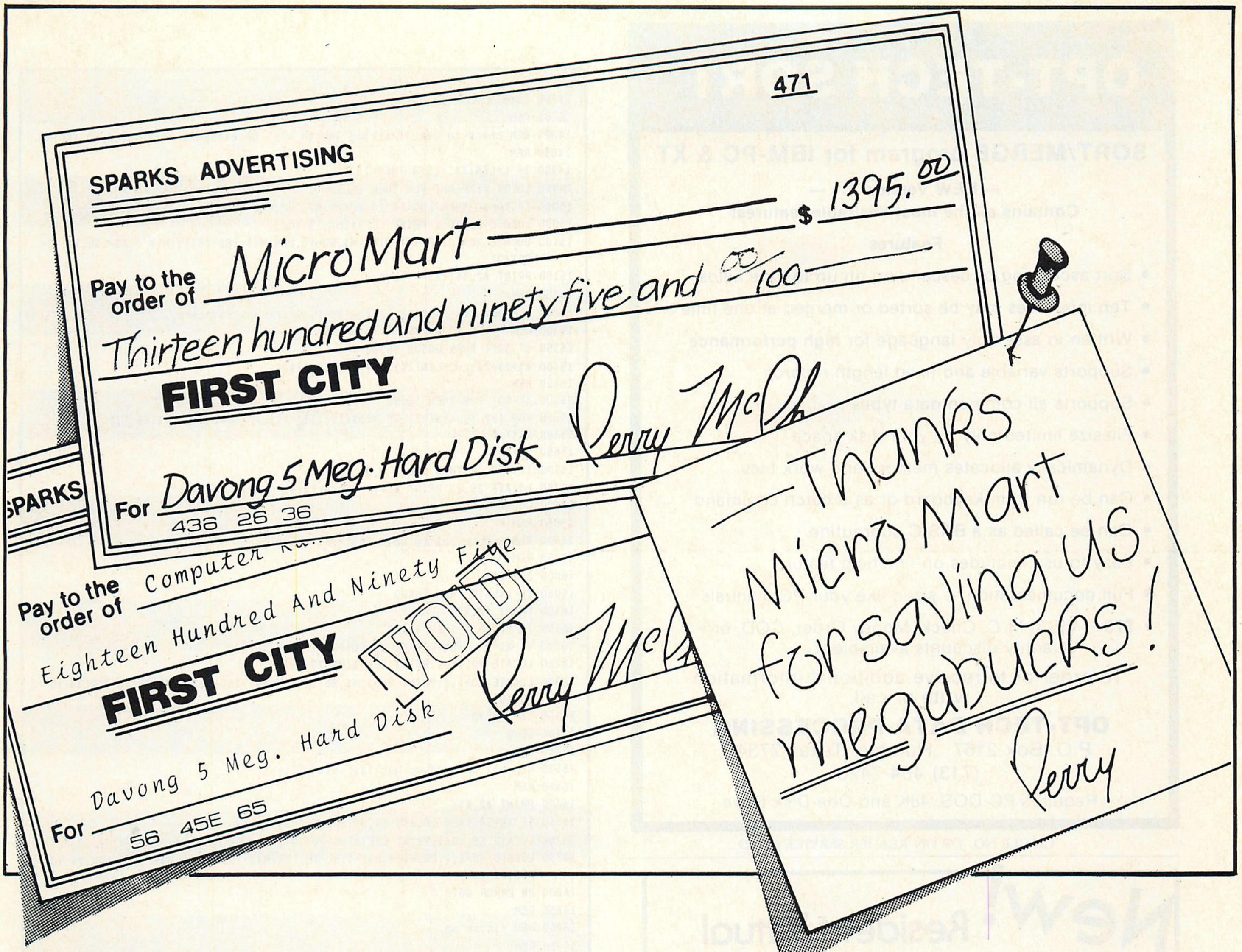
INFORMATION

about the products and services advertised in this issue of PC Tech Journal.

See other side for directions.

TECH PC JOURNAL





before you write that check...

check with us!

Micro Mart means business at a discount. We are the IBM Specialists. Naturally we should be your first choice (why void checks) for supplies in advanced applications for IBM-PCs. You're enhancing your PCs through networking, applying hard discs, setting up communications with mainframes and you have special software needs, so check with us first and you won't have to go anywhere else. Why? Because we deal in state of the art PC related products at the best prices, with the best availability from our Atlanta warehouses. Our present customers include 50 of the top 100 Fortune corporations, multiple IBM-PC user groups and thousands of end-users. So why not let us make the most of your PC budget, too! And by the way, check our ad in this month's PC Magazine and give us a call. Ask for expert advice and your best terms.

MICRO MART

ORDERS ONLY 1-800-241-8149 • For Information 1-404-449-8089

(SEE OUR 2-PAGE AD IN PC MAGAZINE!)

CIRCLE NO. 171 ON READER SERVICE CARD

```
14700 GOTO 12900
14750 REM
14800 REM check to see if string length will be exceeded.  If so, dump string.
14850 REM
14900 IF L+LEN(Z$)<LSTR THEN 15300
14950 COLOR 7:IF NOT FLG THEN 15150
15000 IF XO=24 THEN LOCATE 24,80:PRINT " ";:XO=23
15050 LOCATE XO,YO+1:PRINT LEFT$(M$,79-YO);
15100 LOCATE XO+1,1:PRINT MID$(M$,80-YO,LEN(M$)-(80-YO));:M$="":XO=CSRLIN:YO=
      POS(0)
15150 PRINT #2,X$;:X$="":L=0
15200 REM
15250 REM add test string to the text. Increment counters.
15300 REM
15350 IF SOFT THEN GOSUB 25300
15400 X$=X$+Z$:L=L+LEN(Z$):KK#=KK#+LEN(Z$)
15450 REM
15500 IF NOT FLG THEN 15650
15550 FOR I=1 TO LEN(Z$):IF MID$(Z$,I,1)<>L$ THEN M$=M$+MID$(Z$,I,1)
15600 NEXT I
15650 Z$=""
15700 IF ICOLOR THEN COLOR 3,0
15750 LOCATE 25,20:PRINT KK#;" characters, ";KW#;" words.";
15800 COLOR 7,0
15850 REM
15900 REM update zz$ to hold last character
15950 REM
16000 ZZ$=Q$
16050 IF L<LSTR THEN 16400
16100 PRINT #2,X$;
16150 IF NOT FLG THEN 16350
16200 IF XO=24 THEN LOCATE 24,80:PRINT " ";:XO=23
16250 LOCATE XO,YO+1:PRINT LEFT$(M$,79-YO);
16300 LOCATE XO+1,1:PRINT MID$(M$,80-YO,LEN(M$)-(80-YO));:M$="":XO=CSRLIN:YO=
      POS(0)
16350 X$="":L=0
16400 WEND
16450 REM
16500 REM all done--dump out last bit of text
16550 REM
16600 PRINT #2,X$;
16650 IF XO=24 THEN LOCATE 24,80:PRINT " ";:XO=23
16700 LOCATE XO,YO+1:PRINT LEFT$(M$,79-YO);:IF LEN(M$)<=80-YO THEN 16800
16750 LOCATE XO+1,1:PRINT MID$(M$,80-YO,LEN(M$)-(80-YO));:M$="":XO=CSRLIN:YO=
      POS(0)
16800 ON ERROR GOTO 0
16850 REM
16900 REM finish up.
16950 REM
17000 COLOR 6,0:PRINT:PRINT:PRINT "Processing complete."
17050 COLOR 7,0
17100 PRINT "File ";:COLOR 8,7:PRINT F$;:COLOR 7,0:PRINT
      " has been copied to file ";
17150 COLOR 8,7:PRINT G$;:COLOR 7,0:PRINT " in ";NAM$(TARGET,1);" format."
17200 LOCATE 25,5:PRINT"Done...        "
17250 SOUND 1000,1:SOUND 700,1:SOUND 1800,2:SOUND 800,1
17300 CLOSE
17350 PRINT "Strike any key to return DOS"
17400 A$=INKEY$:IF LEN(A$)=0 THEN 17400
17450 RETURN
17500 REM
17550 REM handle word wrap character
17600 REM
17650 IF SOFT AND LSH(MS)=A-1 THEN MID$(Z$,LSH(MS),1)="-":LSH(MS)=0:MS=MS-1
17700 IF SOFT AND MS=0 THEN SOFT=FALSE
17750 R$="":S$="":ON ERROR GOTO 19300 'in case we read off end...
17800 IF SOURCE<3 THEN 18550
17850 REM
17900 REM read next character; test for e-o-f
17950 REM
18000 R$=INPUT$(1,#1):
18050 IF(EOF(1)) THEN RETURN
18100 REM
18150 REM read next char; test for e-o-f; build test string of at most 4 char
18200 REM
18250 REM
18300 S$=INPUT$(1,#1):IF EOF(1) THEN RETURN ELSE T$=MID$(Z$,A)+R$+S$
18350 REM
18400 REM see if it's a paragraph even though we thought it was a wrap...
18450 REM
18500 IF T$=PAR$(SOURCE) THEN B=A:A=0:GOSUB 19350:Z$=Z$+R$+S$:RETURN
18550 IF RIGHT$(Z$,3)=OVR$(SOURCE) THEN C=LEN(Z$)-NOVR(SOURCE)+1:A=0:GOSUB
      19950:RETURN
18600 REM
18650 REM special case for Volkswriter: chr$(20),cr,lf
18700 REM
18750 IF SOURCE<>2 OR A<2 THEN 19100
18800 REM
```

```
18850 T$=MID$(Z$,A-1,1):IF ASC(T$)<>20 THEN 19050
18900 REM
18950 B=A-1:A=0:GOSUB 19350:RETURN
19000 REM
19050 REM ok, standard wrap situation
19100 REM
19150 Z$=LEFT$(Z$,A-1):Z$=Z$+WRP$(TARGET)+R$+S$
19200 COLOR 2,0:LOCATE 25,60:PRINT "                    ";:LOCATE 25,60:PRINT
      "<Wrap Line>";
19250 A=0
19300 RETURN
19350 REM
19400 IF SOFT AND LSH(MS)=B-1 THEN MID$(Z$,LSH(MS),1)="-":LSH(MS)=0:MS=MS-1
19450 IF SOFT AND MS=0 THEN SOFT=FALSE
19500 REM paragraph handler
19550 REM
19600 Z$=LEFT$(Z$,B-1):Z$=Z$+PAR$(TARGET)
19650 COLOR 6,0:LOCATE 25,60:PRINT "                    ";:LOCATE 25,60:PRINT
      "<Paragraph>";
19700 B=0
19750 RETURN
19800 REM
19850 REM overstrike handler
19900 REM
19950 R$="":S$=""
20000 IF SOURCE<>1 THEN 20100 'only ws indicates overstrike with a single
      cr...
20050 R$=INPUT$(1,#1):IF R$=L$ THEN B=C:C=0:GOSUB 19350:RETURN
20100 Z$=LEFT$(Z$,C-1):Z$=Z$+OVR$(TARGET)
20150 COLOR 5,0:LOCATE 25,60:PRINT "                    ";:LOCATE 25,60:PRINT
      "<Overstrike>";
20200 C=0
20250 RETURN
20300 REM
20350 REM bold handler
20400  REM
20450 BLDFLG=NOT BLDFLG
20500 R$=BLD$(TARGET):IF NOT BLDFLG THEN R$=BLDOFF$(TARGET)
20550 IF D=1 THEN Z$="": GOTO 20650
20600 Z$=LEFT$(Z$,D-1)
20650 Z$=Z$+R$
20700 D=0
20750 RETURN
20800 REM
20850 REM shadow handler
20900 REM
20950 SHDFLG=NOT SHDFLG
21000 R$=SHD$(TARGET):IF NOT SHDFLG THEN R$=SHDOFF$(TARGET)
21050 IF E=1 THEN Z$="":GOTO 21150
21100 Z$=LEFT$(Z$,E-1)
21150 Z$=Z$+R$
21200 E=0
21250 RETURN
21300 REM
21350 REM underline handler
21400 REM
21450 UNDFLG=NOT UNDFLG
21500 R$=UND$(TARGET):IF NOT UNDFLG THEN R$=UNDOFF$(TARGET)
21550 IF F=1 THEN Z$="":GOTO 21650
21600 Z$=LEFT$(Z$,F-1)
21650 Z$=Z$+R$
21700 F=0
21750 RETURN
21800 REM
21850 REM superscript handler
21900 REM
21950 IF SUBFLG AND TARGET=1 THEN R$=SUBOFF$(TARGET):SUBFLG=0:GOTO 22100
22000 SUPFLG=NOT SUPFLG
22050 R$=SUP$(TARGET):IF NOT SUPFLG THEN R$=SUPOFF$(TARGET)
22100 IF G=1 THEN Z$="":GOTO 22200
22150 Z$=LEFT$(Z$,G-1)
22200 Z$=Z$+R$
22250 G=0
22300 RETURN
22350 REM
22400 REM subscript handler
22450 REM
22500 IF SUPFLG AND TARGET=1 THEN R$=SUPOFF$(TARGET):SUPFLG=0:GOTO 23150
22550 SUBFLG=NOT SUBFLG
22600 R$=SUB$(TARGET):IF NOT SUBFLG THEN R$=SUBOFF$(TARGET)
22650 IF H=1 THEN Z$="":GOTO 22750
22700 Z$=LEFT$(Z$,H-1)
22750 Z$=Z$+R$
22800 H=0
22850 RETURN
22900 IF BLDFLG THEN D$=BLDOFF$(SOURCE):RETURN
22950 IF UNDFLG THEN F1$=UNDOFF$(SOURCE):RETURN
23000 IF SHDFLG THEN E$=SHDOFF$(SOURCE):RETURN
```

PULL. PULL. STICK. STICK. CHANGE PERFORMANCE THAT QUICK.

What you see here is a remarkable system that will change the personality of your IBM Personal Computer. A system that makes it possible for you to use your PC to develop applications that normally require the resources of a much bigger computer. A system that quite literally transforms your computer from personal to professional. Here's how:

Easy As A,B,C

First you apply the character labels to your keyboard, then you plug in the custom ROM that will enable your PC to display these new characters. Third, you insert the disk and load the software to complete the change: APL★PLUS®/PC System from STSC.

Now, instead of limiting yourself to off-the-shelf spreadsheets, word processors, or slow program development in BASIC, you can develop custom solutions to problems that previously were beyond the scope of your PC.

The APL★PLUS/PC System is a complete APL language processor from STSC, the leading supplier of APL products and services for more than 14 years. It also includes STSC's business report formatting feature, communications capabilities, and data management features.

The APL★PLUS/PC System can interface to non-APL programs and data files so you can integrate your other software packages with the APL★PLUS/PC. And APL programs developed on your PC will run on other desktop computers, minis, and mainframes that support APL. Software developers may license the APL★PLUS/PC System to develop APL applications for resale using a run-time version of the system.

Everything You Need for Application Development

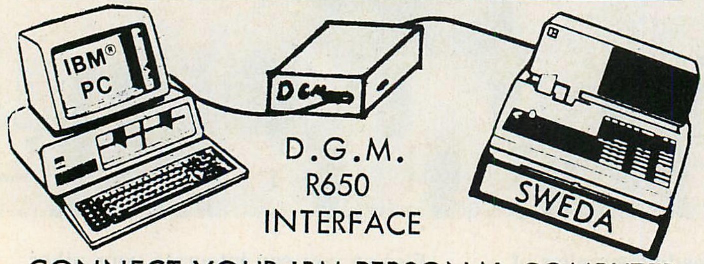
It's all here in our comprehensive package. In addition to the keycap labels, custom ROM, and software disk, you also get a reference manual, an introductory tutorial, and a widely used APL textbook. All for only $595.*

The APL★PLUS/PC System requires an IBM PC or IBM PC XT with PC DOS, at least 128K RAM memory, and one floppy disk drive. To order APL for your PC, call or write STSC, Inc., 2115 East Jefferson Street, Rockville, Maryland 20852, (301) 984-5123.

*U.S.A. price only.

stsc
A Contel Company

CIRCLE NO. 208 ON READER SERVICE CARD



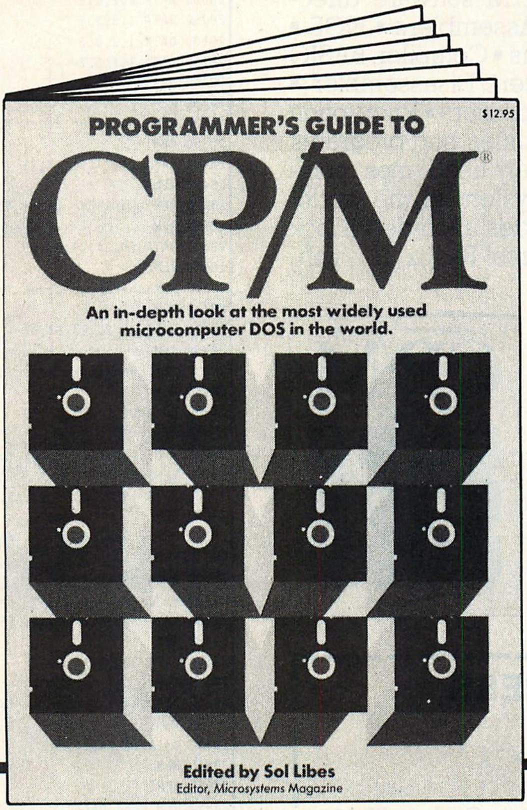

```
23050 IF SUPFLG THEN G1$=SUPOFF$(SOURCE):RETURN
23100 IF SUBFLG THEN H$=SUBOFF$(SOURCE):RETURN
23150 RETURN
23200 REM
23250 LOCATE K,1:COLOR K1:PRINT "Enter the number of characters used to";A$;:
      INPUT NC
23300 IF NC THEN RETURN
23350 IF K<10 AND K1<6 AND NOT MODFY THEN SOUND 200,1:GOTO 23250
23400 RETURN
23450 U$=" ":IF MODFY THEN U$=" (modified)"
23500 COLOR 5,0:PRINT:PRINT "Special Character Definition from File= ";T$;U$
23550 PRINT:PRINT "Function";TAB(15);"`ON' Char";TAB(27);"Dec Ascii";:COLOR 4,
      0:PRINT TAB(46);"`OFF' Char";TAB(57);"Dec Ascii":PRINT
23600 W$=WRP$(IDX)
23650 GOSUB 25100
23700 COLOR 5,0:PRINT "Wrap-around";TAB(20);W$;TAB(27);:FOR I=1 TO LEN(W$):
      PRINT A1(I);:NEXT:PRINT TAB(45)"|"
23750 W$=PAR$(IDX):GOSUB 25100
23800 PRINT "Paragraph";TAB(20);W$;TAB(27);:FOR I=1 TO LEN(W$):PRINT A1(I);:
      NEXT:PRINT TAB(45)"|"
23850 W$=OVR$(IDX):GOSUB 25100
23900 PRINT "Overstrike";TAB(20);W$;TAB(27);:FOR I=1 TO LEN(W$):PRINT A1(I);:
      NEXT:PRINT TAB(45)"|"
23950 W$=BLD$(IDX):GOSUB 25100
24000 PRINT "Bold";TAB(20);W$;TAB(27);:FOR I=1 TO LEN(W$):PRINT A1(I);:NEXT
24050 W$=BLDOFF$(IDX):GOSUB 25100:PRINT TAB(45)"|";
24100 COLOR 4:PRINT TAB(50);W$;TAB(57);:FOR I=1 TO LEN(W$)-1:PRINT A1(I);:NEXT:
      PRINT A1(I)
24150 W$=UND$(IDX):GOSUB 25100
24200 COLOR 5:PRINT "Underscore";TAB(20);W$;TAB(27);:FOR I=1 TO LEN(W$):PRINT
      A1(I);:NEXT
24250 W$=UNDOFF$(IDX):GOSUB 25100:PRINT TAB(45)"|";
24300 COLOR 4:PRINT TAB(50);W$;TAB(57);:FOR I=1 TO LEN(W$)-1:PRINT A1(I);:NEXT:
      PRINT A1(I)
24350 W$=SHD$(IDX):GOSUB 25100
24400 COLOR 5:PRINT "Shadow";TAB(20);W$;TAB(27);:FOR I=1 TO LEN(W$):PRINT A1(I)
      ;:NEXT
24450 W$=SHDOFF$(IDX):GOSUB 25100:PRINT TAB(45)"|";
24500 COLOR 4:PRINT TAB(50);W$;TAB(57);:FOR I=1 TO LEN(W$)-1:PRINT A1(I);:NEXT:
      PRINT A1(I)
24550 W$=SUP$(IDX):GOSUB 25100
24600 COLOR 5:PRINT "Superscript";TAB(20);W$;TAB(27);:FOR I=1 TO LEN(W$):PRINT
      A1(I);:NEXT
24650 W$=SUPOFF$(IDX):GOSUB 25100:PRINT TAB(45)"|";
24700 COLOR 4:PRINT TAB(50);W$;TAB(57);:FOR I=1 TO LEN(W$)-1:PRINT A1(I);:NEXT:
      PRINT A1(I)
24750 W$=SUB$(IDX):GOSUB 25100
24800 COLOR 5:PRINT "Subscript";TAB(20);W$;TAB(27);:FOR I=1 TO LEN(W$):PRINT
      A1(I);:NEXT
24850 W$=SUBOFF$(IDX):GOSUB 25100:PRINT TAB(45)"|";
24900 COLOR 4:PRINT TAB(50);W$;TAB(57);:FOR I=1 TO LEN(W$)-1:PRINT A1(I);:NEXT:
      PRINT A1(I)
24950 PRINT:PRINT "Strike any key to continue"
25000 W$=INKEY$:IF LEN(W$)=0 THEN 25000
25050 RETURN
25100 FOR I=1 TO LEN(W$):A1(I)=ASC(MID$(W$,I)):IF A1(I)=13 THEN MID$(W$,I)=CHR$
      (155) ELSE IF A1(I)=10 THEN MID$(W$,I)=CHR$(192)
25150 NEXT I
25200 RETURN
25250 REM
25300 REM
25350 FOR M=1 TO MS
25400 IF LSH(M)=0 THEN 25600
25450 Z$=LEFT$(Z$,LSH(M)-1)+MID$(Z$,LSH(M)+1)
25500 LSH(M)=0
25550 FOR K9=M+1 TO MS:LSH(K9)=LSH(K9)-1:NEXT K9
25600 NEXT M
25650 SOFT=FALSE:MS=0
25700 RETURN
25750 REM
25800 REM initialization routine
25850 REM
25900 REM
25950 REM
26000 REM processor names
26050 REM
26100 DATA WORDSTAR,VOLKSWRITER,EASYWRITER,EDLIN,USER1,USER2
26150 DATA WS,VW,EW,ED,US1,US2
26200 REM
26250 REM number of wrap,paragraph,overstrike,bold,and underscore characters
26300 REM
26350 DATA 2,2,2,2
26400 DATA 2,3,4,4
26450 DATA 1,3,0,0
26500 DATA 1,2,0,2
26550 DATA 1,3,0,3
26600 REM
26650 REM wrap characters
```

PROGRAMMER'S GUIDE TO CP/M®

Edited by Sol Libes

Here's an important collection of CP/M insights that you'll never find in any CP/M manual. CP/M is the most popular microcomputer DOS in use today, and this widespread use has generated many innovative techniques and enhancements of CP/M. *Programmer's Guide to CP/M* tells you what these enhancements are and how to put them to use, how to get around apparent limitations of a CP/M system and why CP/M is far more versatile than you might have imagined. Every article in *Programmer's Guide to CP/M* originally appeared in MICROSYSTEMS between January 1980 and February 1982. Except for this collection, these articles are now unavailable! *Programmer's Guide to CP/M* gives you an in-depth look at CP/M from the viewpoint of the programmer—the individual who creates the software that interfaces directly with CP/M, or who is installing CP/M on systems for which configurations do not already exist.

Contents include "An Introduction to CP/M," "The CP/M Connection," "CP/M Software Reviews," "CP/M Utilities & Enhancement," "CP/M 86" and "CP/M Software Directories." $12.95.



Also available at your local bookstore or computer store.

For faster service, PHONE TOLL FREE
800-631-8112
(In NJ only: 201-540-0445)

CP/M is a registered trademark of Digital Research, Inc.

NF5F

MICROSYSTEMS PRESS

Dept. NF5F · 39 East Hanover Avenue · Morris Plains, NJ 07950

Please send me _____ *Programmer's Guide to CP/M* at $12.95* plus $2.00 postage and handling each. Outside USA add $3.00 per order. **#14C**

☐ **PAYMENT ENCLOSED** $_______. *Residents of CA, NJ and NY State add applicable sales tax.

☐ **CHARGE MY:**
(Charge and phone orders $10 minimum.)
☐ American Express
☐ MasterCard ☐ Visa

Card No.____________________ Exp. Date________

Signature____________________

Mr./Mrs./Ms.____________________
(please print full name)

Print Name____________________

Address____________________ Apt.______

City____________________

State____________ Zip____________

☐ Send me a FREE *Creative Computing* Catalog.

```
26700 REM
26750 DATA 141,10,13,10,13,10,13,10
26800 REM
26850 REM paragraph characters
26900 REM
26950 DATA 13,10,20,13,10,13,10,13,10,13,10,13,10
27000 REM
27050 REM overstrike characters
27100 REM
27150 DATA 13,13,13,10
27200 REM
27250 REM bold characters
27300 REM
27350 DATA 02,27,69,27,69
27400 REM
27450 REM underscore characters
27500 REM
27550 DATA 19,27,45,01,27,45,01
27600 REM
27650 REM
27700 REM number of shadow,superscript,subscript,bold-off,underscore-off,
27750 REM           shadow-off,super-off,sub-off characters
27800 REM
27850 DATA 1,2,0,2
27900 DATA 1,1,0,3
27950 DATA 1,1,0,3
28000 DATA 1,2,0,2
28050 DATA 1,3,0,3
28100 DATA 1,2,0,2
28150 DATA 1,1,0,2
28200 DATA 1,1,0,2
28250 REM
28300 REM character codes for shadow text
28350 REM
28400 DATA 04,27,71,27,71
28450 REM
28500 REM character codes for superscripts
28550 REM
28600 DATA 20,24,27,83,00
28650 REM
28700 REM character codes for subscripts
28750 REM
28800 DATA 22,25,27,83,01
28850 REM
28900 REM character codes for turning bold off
28950 REM
29000 DATA 02,27,70,27,70
29050 REM
29100 REM character codes for turning understrike off
29150 REM
29200 DATA 19,27,45,0,27,45,0
29250 REM
29300 REM character codes for turning shadow off
29350 REM
29400 DATA 04,27,72,27,72
29450 REM
29500 REM character codes for turning superscripts off
29550 REM
29600 DATA 20,25,27,84
29650 REM
29700 REM character codes for turning subscripts off
29750 REM
29800 DATA 22,24,27,84
29850 REM
29900 REM
29950 REM read data
30000 REM
30050 REM names of processors
30100 REM
30150 RESTORE 26100:FOR J=1 TO 2:FOR I=1 TO N:READ NAM$(I,J):NEXT I:NEXT J
30200 RESTORE 26350:FOR I=1 TO 4: READ NWRP(I):NEXT I 'no. wrap chars
30250 RESTORE 26400:FOR I=1 TO 4: READ NPAR(I):NEXT I 'no. paragraph chars
30300 RESTORE 26450:FOR I=1 TO 4: READ NOVR(I):NEXT I 'no. overstrike chars
30350 RESTORE 26500:FOR I=1 TO 4: READ NBLD(I):NEXT I 'no. bold characters
30400 RESTORE 26550:FOR I=1 TO 4: READ NUND(I):NEXT I 'no. understrike chars
30450 REM
30500 REM read wrap,paragraph,overstrike,bold,and underscore chars
30550 REM
30600 REM
30650 REM wrap
30700 REM
30750 RESTORE 26750:FOR I=1 TO 4:FOR J=1 TO NWRP(I):READ A:WRP$(I)=WRP$(I)+CHR$
      (A):NEXT J:NEXT I
30800 REM
30850 REM paragraph
30900 REM
30950 RESTORE 26950:FOR I=1 TO 4:FOR J=1 TO NPAR(I):READ A:PAR$(I)=PAR$(I)+CHR$
      (A):NEXT J:NEXT I
```

```
31000 REM
31050 REM overstrike
31100 REM
31150 RESTORE 27150:FOR I=1 TO 4
31200 IF NOVR(I)=0 THEN 31300
31250 FOR J=1 TO NOVR(I):READ A:OVR$(I)=OVR$(I)+CHR$(A):NEXT J
31300 NEXT I
31350 REM
31400 REM bold
31450 REM
31500 RESTORE 27350:FOR I=1 TO 4
31550 IF NBLD(I)=0 THEN 31650
31600 FOR J=1 TO NBLD(I):READ A:BLD$(I)=BLD$(I)+CHR$(A):NEXT J
31650 NEXT I
31700 REM
31750 REM understrike
31800 REM
31850 RESTORE 27550:FOR I=1 TO 4
31900 IF NUND(I)=0 THEN 32000
31950 FOR J=1 TO NUND(I):READ A:UND$(I)=UND$(I)+CHR$(A):NEXT J
32000 NEXT I
32050 REM
32100 REM number of shadow,super,sub,bold-off,understrike-off chars
32150 REM
32200 RESTORE 27850:FOR I=1 TO 4:READ NSHD(I):NEXT I
32250 RESTORE 27900:FOR I=1 TO 4:READ NSUP(I):NEXT I
32300 RESTORE 27950:FOR I=1 TO 4:READ NSUB(I):NEXT I
32350 RESTORE 28000:FOR I=1 TO 4:READ NBLDO(I):NEXT I
32400 RESTORE 28050:FOR I=1 TO 4:READ NUNDO(I):NEXT I
32450 RESTORE 28100:FOR I=1 TO 4:READ NSHDO(I):NEXT I
32500 RESTORE 28150:FOR I=1 TO 4:READ NSUPO(I):NEXT I
32550 RESTORE 28200:FOR I=1 TO 4:READ NSUBO(I):NEXT I
32600 REM
32650 REM shadow characters
32700 REM
32750 RESTORE 28400:FOR I=1 TO 4
32800 IF NSHD(I)=0 THEN 32900
32850 FOR J=1 TO NSHD(I):READ A:SHD$(I)=SHD$(I)+CHR$(A):NEXT J
32900 NEXT I
32950 REM
33000 REM superscript
33050 REM
33100 RESTORE 28600:FOR I=1 TO 4:
33150 IF NSUP(I)=0 THEN 33250
33200 FOR J=1 10 NSUP(I):READ A:SUP$(I)=SUP$(I)+CHR$(A):NEXT J
33250 NEXT I
33300 REM
33350 REM subscripts
33400 REM
33450 RESTORE 28800:FOR I=1 TO 4:
33500 IF NSUB(I)=0 THEN 33600
33550 FOR J=1 TO NSUB(I):READ A:SUB$(I)=SUB$(I)+CHR$(A):NEXT J
33600 NEXT I
33650 REM
33700 REM bold-off
33750 REM
33800 RESTORE 29000:FOR I=1 TO 4:
33850 IF NBLDO(I)=0 THEN 33950
33900 FOR J=1 TO NBLDO(I):READ A:BLDOFF$(I)=BLDOFF$(I)+CHR$(A):NEXT J
33950 NEXT I
34000 REM
34050 REM understrike off
34100 REM
34150 RESTORE 29200:FOR I=1 TO 4:
34200 IF NUNDO(I)=0 THEN 34300
34250 FOR J=1 TO NUNDO(I):READ A:UNDOFF$(I)=UNDOFF$(I)+CHR$(A):NEXT J
34300 NEXT I
34350 REM
34400 REM shadow
34450 REM
34500 RESTORE 29400:FOR I=1 TO 4:
34550 IF NSHDO(I)=0 THEN 34650
34600 FOR J=1 TO NSHDO(I):READ A:SHDOFF$(I)=SHDOFF$(I)+CHR$(A):NEXT J
34650 NEXT I
34700 REM
34750 REM superscript-off
34800 REM
34850 RESTORE 29600:FOR I=1 TO 4:
34900 IF NSUPO(I)=0 THEN 35000
34950 FOR J=1 TO NSUPO(I):READ A:SUPOFF$(I)=SUPOFF$(I)+CHR$(A):NEXT J
35000 NEXT I
35050 REM
35100 REM subscript off
35150 REM
35200 RESTORE 29800:FOR I=1 TO 4:
35250 IF NSUBO(I)=0 THEN 35350
35300 FOR J=1 TO NSUBO(I):READ A:SUBOFF$(I)=SUBOFF$(I)+CHR$(A):NEXT J
35350 NEXT I
35400 RETURN
```

Two Approaches To Pascal

If you're interested in Pascal, you should know that all Pascals are not alike. The Pascal book for you depends on the Pascal you want to learn.

Programming the IBM Personal Computer: VCSD Pascal
Seymour H. Pollack
(Holt, Rinehart and Winston, New York, 1983)
323 pages, paperback, $17.95

Introduction to Programming with ESP and Pascal
Allen B. Tucker, Jr.
(Holt, Rinehart and Winston, New York, 1983)
362 pages, paperback, $19.95

ESP? What does ESP have to do with microcomputers? Come to think of it, what does "PC" stand for, anyway? Precognition and Computing? Prophets and Conjuring? Preternatural Contacts? Personal Crystal-gazing? When I received the book *Introduction to Programming with ESP and Pascal,* I began to wonder. A glance at the book's preface, however, soon brought me back to the earthly plane: ESP is short for "Eight Statement Pascal," the subset of Pascal the author has chosen to present first. This is one of two new books that may be useful to new users of UCSD Pascal on the PC.

Programming the IBM Personal Computer: UCSD Pascal is specifically for new users of UCSD Pascal on the IBM PC and includes a chapter acquainting the newcomer with the UCSD p-System. It succeeds in presenting the material in a useful, interesting way and has exercises at the end of each chapter. Some are pencil-and-paper exercises (alas, no solutions provided), and some are programming assignments.

A beginner can't expect to understand the full uses of the more advanced features the first time through.

This book presents Pascal in an orderly fashion. The first four chapters give the reader a general feel for the language and are very well done. Each subsequent chapter is dedicated to one major area of Pascal, making the book a useful reference for later. (In books of this sort, I've noticed that reference is as important to me as the first reading.)

The second book, *Introduction to Programming with ESP and Pascal,* was written as a text for an introductory college-level course. Its most outstanding feature is its "lab" projects: 24 small programming assignments, which allow students to practice what they've learned at regular intervals. But that's not all—each lab is really three assignments in one. One is business-oriented, one emphasizes math and science, and the third is of more general interest. Each of the three assignments in the group teaches the same concept. Along with the labs, exercises are provided at frequent intervals, with answers to some in the back of the book.

ESP and Pascal introduces new ideas only when they are needed, so newcomers to Pascal can pick up the more difficult concepts as they go along. The eight simple statements of Eight Statement Pascal (ESP) are introduced first, but the author doesn't overdo the simplification. When more statements are needed, they are promptly introduced. There's a disadvantage, however; by sacrificing organization, the author makes it difficult for the reader to achieve an overview of all the material or to locate a particular section again later.

Unfortunately, *Introduction to Programming with ESP and Pascal* is oriented toward a college mainframe environment. It does contain a section on UCSD Pascal's STRINGS, but most of the examples are written in Waterloo Pascal. Consequently, some sample programs won't run properly on UCSD Pascal. For instance, one of the samples relied on "McCarthy eval-

uation," where the second operand of an And/Or statement is only evaluated when necessary. UCSD Pascal doesn't provide this feature, so the program bombs; no hint of this problem is given.

In Contrast. . .

Programming the IBM PC: UCSD Pascal doesn't suffer from these problems. Its advantages over *ESP and Pascal* are best seen by looking at two specific areas: data types and input/output. *ESP and Pascal* introduces the different data types in a number of places. Predefined types are presented in chapter 2; type declarations are introduced briefly in chapter 3 (unfortunately, one of the two examples is totally wrong); record types are taught in chapter 8; and the remainder are dealt within chapter 11, squeezed in with sets. *Programming the IBM PC: UCSD Pascal,* on the other hand, introduces predefined types in chapter 3, but thoroughly teaches all types in chapter, titled "Data." A beginner can't expect to understand the full uses of the more advanced features the first time through, but I prefer such an approach because it allows the student to choose how much he would like to learn about a particular subject and gives a better idea of what is still to be learned.

Input/output is always a thorny problem in Pascal. Different Pascals handle interactive input differently. *ESP and Pascal* doesn't deal with this at all; it provides input/output in a more informative way. Because it was designed with the p-System in mind, it gives the correct methods.

To sum up, *Programming the IBM PC: UCSD Pascal* is a good choice for beginners who wish to learn to program using the PC's UCSD Pascal. The book has a pleasant style, and is easy enough to understand that a beginner can learn the language with little or no extra help. Its excellent organization makes it valuable as an understandable reference that can be used time after time—a real winner.

Introduction to Programming with ESP and Pascal is less useful for someone trying to learn UCSD Pascal from scratch—its pitfalls require occasional help from someone with experience with UCSD Pascal. Unfortunately, it's easy to tell that this book is intended as a text. It reads like one, and can be a bit heavy at times. Its excellent projects and exercises are useful, though, and its later chapters do a good job of describing advanced data structures. I would recommend this book to anyone who wants plenty of projects in different subject areas and who can get a friend to help him or her past the difficulties of conversion to a different version of Pascal. It is also a good choice for users with some Pascal experience who would like to brush up their skills, or as a supplementary text for a beginning Pascal course. Marc Ringuette

Where Am I?

How to find out "where you are" in DOS 2.0's hierarchical directory structure

WILL FASTIE

One of the most important features of DOS 2.0, especially to software developers, is its hierarchical directory structure. This structure allows organization of files in directories, with the hope that the management of large numbers of files thereby is simplified.

Whether your system uses diskette or hard disk drives, the potential for confusion is real. In my specific case, the XT's hard disk contains 28 directories and 362 files (about 5 megabytes worth). Although it is unlikely that a single floppy would have that complexity, a set of diskettes certainly could.

Much like DOS 1.1's default drive (e.g., A:, B:, etc.), DOS 2.0 allows a default directory. The confusion comes from not knowing which directory is the current default. The problem is that DOS provides no direct way to query for the current directory name.

The command DIR does display the name of directory before it lists the files, but this is not particularly neat. If the list of files is long, the directory name will scroll away. Avoid this by depressing the BREAK key. Again, it's not neat.

THE SOLUTION

There is a way to have DOS 2.0 *passively* state the current directory name. It is accomplished with the PROMPT command, which can alter the DOS prompt from the traditional drive letter (e.g., C:) to just about anything. In addition, a number of system parameters may be displayed, including the time, date, DOS version number, the default drive, and the default directory on the default drive.

To solve my problem, I modified my AUTOEXEC.BAT file to contain the command

PROMPT [$p]$g

which produces DOS prompts of the form

[C:/copy 102/fastie]⟩

In this way, the name of the current directory can be seen whenever the system is at the DOS command level, and the effect of a change directory command (CHDIR or CD) can be seen instantly.

THE PROMPT COMMAND

The PROMPT command uses a small "meta-language" which describes the desired prompt string to DOS. Any ASCII character can be placed in the prompt. Preceding certain characters with the dollar sign ($) causes them to be interpreted by the PROMPT command. The complete list of characters which the interpreter understands appears in figure 1. Some of the characters in the list (i.e., g, l, b, and q) are included because they have special meaning to the DOS command interpreter; if not prefixed with $ they will have their normal meaning.

In the case above, the square brackets ([and]) are interpreted as ASCII characters and printed as such in the prompt. The two $ commands are $p, ;which is the current directory, and $g, which is the right angle bracket (⟩). The choice of this format was nothing more than a personal one, the meta-language is flexible enough to allow almost any prompt. Some examples are shown in figure 2, a log of a wild session of prompt changing.

Figure 1: List of PROMPT Meta-commmands

Each command shown must be prefaced with the dollar sign ($). Any character used as a command that is not in this list is ignored and no action is taken by the PROMPT command.

Command	*Displays*
$	**dollar sign ($)**
t	**system time**
d	**system date**
n	**default drive identifier**
p	**the default directory (entire path name) on the default drive**
v	**DOS version number**
g	**right angle bracket (⟩)**
l	**left angel bracket (⟨)**
b	**vertical bar (:)**
q	**equal sign (=)**
h	**generates a backspace which will erase the previous character**
e	**ESCape character (ASCII code 27)**
-	**(underscore character) generates a carriage return/line feed (CR/LF) sequence which moves the cursor to the next line on the display**

Figure 2: Examples of Prompt Commands and Results

```
C>prompt [$p]$g
[C:\util]>prompt $v on $d at $t $_$g
IBM Personal Computer DOS Version  2.00 on Wed  7-06-1983 at 14:36:09.35
>prompt $t
14:36:09.95prompt $t$g
14:36:10.50>.Next prompt includes space at end to force space after >
14:36:11.22>prompt $t$g
14:36:11.77> prompt $t$h$h$h$g
14:36:12>prompt [{($l$q$b$q$g)}] ...
[{(<=|=>)}] ...prompt
C>
```

TECH RELEASES

PC software, periferals, expansion cards and new developments

Digital Research and PC-DOS

Digital Research has announced plans to support the PC-DOS operating system with its family of programming languages, thereby making available to PC owners a large number of CP/M-based applications that will run under PC-DOS.

Products that will be available and their prices are these: C-BASIC Compiler™, $600; Pascal MT+™, $600; PC/I™, $750; C $600; LEVEL II Cobol, $1,600; Access Manager™, $400; Display Manager™, $500; Programmer Utilities, $200; and Symbolic Debugger, $150.

Digital Research
P.O. Box 579
Pacific Grove, CA 93950
800-772-3545 Ext. 400
(California)
800-227-1617 Ext. 400
(Nationwide)

CIRCLE NO. 398 ON READER SERVICE CARD

Intellicom™ from Computer Toolbox

Intellicom, an intelligent communications software package, is now available from Computer Toolbox Inc. It provides intelligent computer to computer communications from both a terminal emulation and file transfer standpoint. Intellicom supercedes STERM, a smart terminal emulator, which has been on the market for more than a year.

Intellicom is initially offered for the DEC VT180, IBM PC, NorthStar Advantage and NorthStar Horizon. The price of the IBM PC version $49.95.

Computer Toolbox
1325 East Main Street
Waterbury, CT 06705
203-754-4197

CIRCLE NO. 390 ON READER SERVICE CARD

Plug-In Modems for the PC from SSM

Two direct-connect, auto-dial/auto-answer modems specifically for the PC are available from SSM Microcomputer Products, Inc. The 300-baud PC ModemCard 300 and the 1200-baud PC ModemCard 1200 are designed for unattended dialing and answering. They are Bell 103- and 212A-compatible for full duplex operation at standard 300 or 1200 baud rate. Both include a simple terminal program and a subscription to The Source.

Retail price for the PC ModemCards are $349 for 300 baud and $599 for 1200 baud. The plug-on upgrade card is $329.

SSM Microcomputer
Products, Inc.
2190 Paragon Drive
San Jose, CA 95131
408-946-7400

CIRCLE NO. 382 ON READER SERVICE CARD

RX-80 Printer from Epson

A new high-speed printer is available from Epson, the RX-80, dot matrix, 80-column printer. It delivers a printing speed of 100 characters per second with a choice of two full 96-character ASCII sets plus nine international sets and 32 HX-20 graphics characters.

The RX-80 has graphics capabilities with bit-image resolution to 240 dots per inch horizontally and 216 DPI vertically. The new printer retails for $499.

Epson America Inc.
3415 Kashiwa St.
Torrance, CA 90505

CIRCLE NO. 380 ON READER SERVICE CARD

MicroRIM and Nestar

MicroRim, a producer of relational database software, will provide MicroRIM™, its fully-relational database, in multi-user environments for Nestar's PLAN 4000 system. The two companies signed an OEM agreement, marking MicroRIM's entry into the local area network (LAN) industry.

Nestar's LAN currently links the IBM PC and the Apple II and III. MicroRIM runs under the CP/M and MS-DOS operating systems and provides multiuser capability under MS-DOS by utilizing Nestar's file service.

Nestar will distribute the PLAN 4000 System with MicroRIM through dealers and distributors and directly through national accounts.

Microrim Inc.
1750 112th Ave., NE
Bellevue, WA 98004
206-453-6017
Nestar Systems, Inc.
2585 East Bayshore Road
Palo Alto, CA 94303
415-493-2223

CIRCLE NO. 397 ON READER SERVICE CARD

MBASIC-to-OASIS Basic

MTRANS, a utility that converts Microsoft BASIC programs to executable OASIS BASIC, is now available from Phase One Systems. A thorough, two-pass converter, MTRANS converts any MBASIC program that has been saved in ASCII format and moved to OASIS via the GETFILE command. It displays each line of program text on the CRT screen as it is being converted and informs the operator of any difficulty or errors in a brief report that may be written to a sequential file or printed on an attached printer.

The suggested retail price for the 8-bit version of MTRANS is $245; the 16-bit version is $345.

Phase One Systems, Inc.
7700 Edgewater Drive,
Suite 830
Oakland, CA 94621
415-562-8085

CIRCLE NO. 379 ON READER SERVICE CARD

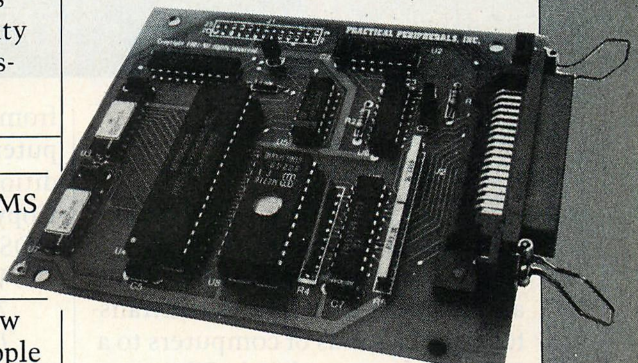
File Transfer Programs from Personal Computer Products

File transfer programs that allow the transfer of files from the Apple II, II+, and IIe, and from Radio Shack Model II, 4, 12, 16 to the IBM PC or XT have been released by Personal Computer Products. Included in the new products is a communications program for the IBM PC as well as an adapter that allows the systems to connect. Each program contains several useful utilities that assist in program conversion and can aid in the preparation of files for editing.

The Apple package is available from Personal Computer Products for $94.95; the Radio Shack package is $89.95.

Personal Computer Products
1400 Coleman Avenue,
Suite C-18
Santa Clara, CA 95050
408-988-0164

CIRCLE NO. 383 ON READER SERVICE CARD



Practical Peripherals Interface Boards

Serial and parallel interfaces, which allow for simultaneous printouts and computer input for most Epson printers, are available from Practical Peripherals, Inc. The boards are Microbuffert™ print buffers MBS-8K, MBP-16K, and MBP-32K.

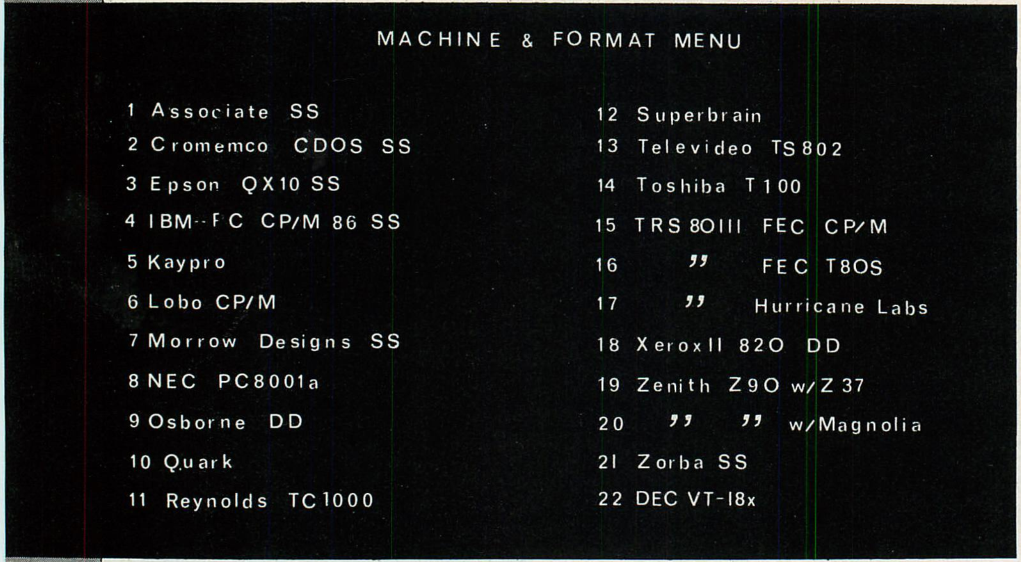
Retail prices of the interface boards range from $299-$349.

Practical Peripherals, Inc.
31245 La Baya Drive
Westlake Village, CA 91362
213-991-8200

CIRCLE NO. 387 ON READER SERVICE CARD

TECH RELEASES

PC software, periferals, expansion cards and new developments



Xeno-Copy Direct File Transfer

Vertex Systems has introduced Xeno-Copy, a software utility that allows direct disk-to-disk file transfer from 22 kinds of computers to a PC-DOS formatted disk. Files can be transferred to the IBM PC or XT from disks formatted in other computers without any modems or additional hardware required. Xeno-Copy is fully compatible with PC DOS 1.1 and 2.0; it sells for $99.50.

Vertex Systems
7950 W. 4th Street
Los Angeles, CA 90048
213-938-0857

CIRCLE NO. 396 ON READER SERVICE CARD

SmarTerm™/PC from PerSoft, Inc.

An addition to Persoft, Inc.'s SmarTerm/PC line of full-featured software terminal emulators for the IBM PC allows an IBM PC or XT system to function as a Data General Corporation D100, D200, or D400 terminal and to transfer program and data files between the PC and host computer system at speeds up to 9600 baud. The model is TE400-FT, and it implements virtually all features of these terminals, including multiple display windows with independent vertical scrolling and window erase, full character display attributes, and local printer support, including "pass thru" mode.

Persoft has also recently released version 2.1 of SmarTerm/PC Model 100-FT, its Digital Equipment Corporation VT100, VT101, VT102, and VT52 terminal emulator/file transfer package.

The new version of SmarTerm/PC is available for both the PC and the XT for $150.

Persoft, Inc.
2470 Ski Lane
Madison, WI 53713
608-233-1000

CIRCLE NO. 388 ON READER SERVICE CARD

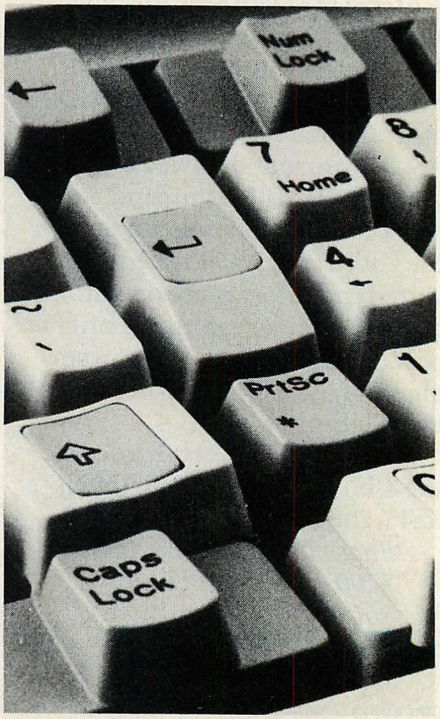
KEYFIXER™ from Vertex

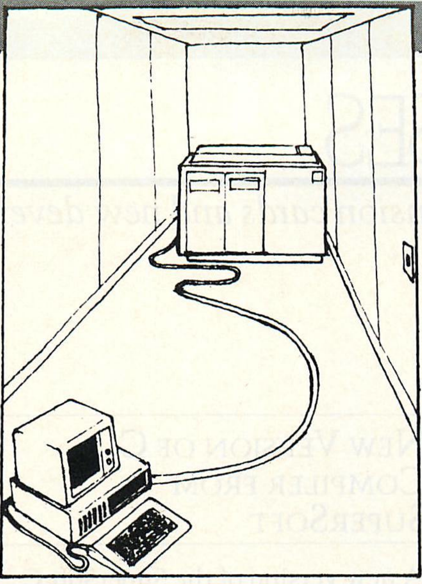
A set of four precision-molded plastic collars, KEYFIXER for the IBM PC and XT, improves typing ease and accuracy by increasing the size of critical typing keys by increasing the fingertip strike area. Similar in color and texture to the original keycaps, they are easily installed and do not permanently alter the keyboard.

KEYFIXERS are supplied for RETURN, LEFT SHIFT, RIGHT SHIFT, and TAB keys. The TAB KEYFIXER can be installed on the BACKSPACE instead.

The set of four is $19.95
Vertex Systems
7950 W. Fourth Street
Los Angeles, CA 90048
213-938-0857

CIRCLE NO. 392 ON READER SERVICE CARD





Digital Research and VisiON™

Digital Research will support VisiCorp's VisiON operating environment as a standard for its CP/M™ operating system products, and will work with original equipment manufacturers of personal computers to adapt VisiON software for CP/M systems.

VisiON is an interactive, graphics-oriented software environment that allows a personal computer user to integrate any number of VisiON and third-party applications programs on the system.

VisiCorp
2895 Zanker Road
San Jose, CA 95134
408-946-9000

CIRCLE NO. 393 ON READER SERVICE CARD

Informatics™ and VisiCorp™

Informatics General Corporation and VisiCorp have announced that they are developing and will market a series of advanced database extraction and communications software products to link personal computers with large IBM mainframe systems. Designed primarily for use in medium to large corporations, the new software will give PC users access to corporate information maintained in centralized databases while providing data processing efficiency, security, and access control needed by corporate management information systems (MIS) directors and data processing managers.

The new products include two cooperating software packages: VisiAnswer™, which operates on the IBM PC, and Answer™/DB, which runs on the IBM mainframe. Informatics/VisiCorp products will be marketed by both companies.

Informatics General Corp.
21031 Vantura Blvd.
Woodland Hills, CA 91364
213-887-9040
VisiCorp
2895 Zanker Road
San Jose, CA 95134
408-942-6081

CIRCLE NO. 394 ON READER SERVICE CARD

PC Modem Speed Adapter™

VEN-TEL Inc. has introduced the PC Modem Speed Adapter™ which can increase the baud rate of the company's PC Modem Plus™ to 1200 baud. The combination is designed specifically for the IBM PC and the PC-XT.

The PC Modem Plus system includes a card-type modem, Crosstalk-XVI™ communications software, a phone cable, and complete instructions. The $360 adapter attaches to the modem in the same slot. All 300 baud PC Modem Plus systems in the field can be upgraded to full 300/1200 baud operation.

VEN-TEL INC.
2342 Walsh Avenue
Santa Clara, CA 95051
408-727-5721

CIRCLE NO. 395 ON READER SERVICE CARD

VRdata's Hard Disk III

At prices that start at $995 for 5 meg, $1249 for 10 meg, and $1449 for 15 meg, VRdata is now offering its Hard Disk III Winchester subsystem, which includes case, power supply, cables, and mounting hardware fully assembled, tested, and burned-in. Adapter kits, which include software, are $150 additional.

VRdata
777 Henderson Blvd.
Folcroft, PA 19032
800-345-8102

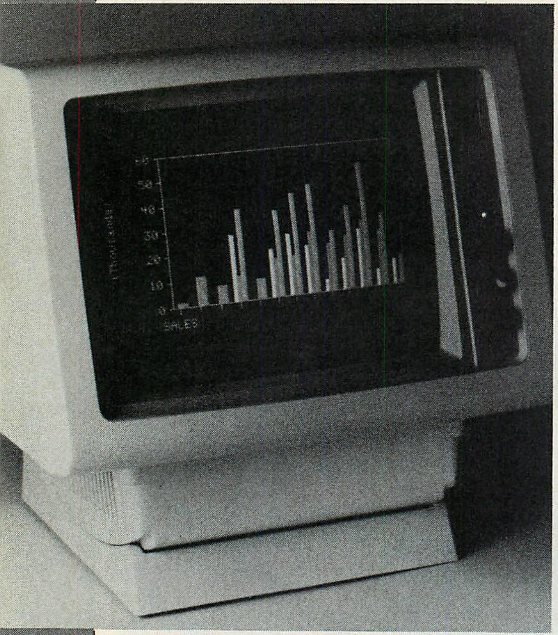
CIRCLE NO. 389 ON READER SERVICE CARD

TECH RELEASES

PC software, periferals, expansion cards and new developments

Pedestal II and the AD-1 Adapter from Curtis

Curtis Manufacturing Company is now producing PC Pedestal II, a tilt and swivel add-on custom base for the IBM PC Color Display, and the AD-1 adapter that can be used with the pedestal to accommodate Princeton Graphics Systems displays and Quadram's Quadrachrome displays.



Pedestal II joins Pedestal I, which was designed for the PC Monochrome Display. For optimal viewing, the pedestals can be tilted or swiveled in any direction to improve sight and minimize glare. Both pedestals are available from computer dealers nationwide for $79.95. The AD-1 adapter is available for $9.95.

Curtis Manufacturing Co., Inc.
Grove Street
Peterborough, NH 03458
603-924-7803

CIRCLE NO. 384 ON READER SERVICE CARD

New Version of C Compiler from SuperSoft

A new version of the SuperSoft C Compiler for the CP/M-80, CP/M-86, and MS DOS operating systems has been released. It is syntactically compatible with UNIX version 7 C, and supports such features as long integer functions and double floating point functions, including trigonometric functions.

Retail price of the compiler is $275 for the CP/M-80 operating system and $500 for other operating systems.

Supersoft
1713 S. Neil St.
Box 1628
Champaign, IL 61820
217-359-2112

CIRCLE NO. 391 ON READER SERVICE CARD

Printer Buffer from von Leivendyke

The Graphics ConvertaBuffer has been announced by von Leivendyke Enterprises. It allows the user to print business graphs generated by programs like 1-2-3 and MBA on letter-quality printers.

The Graphics ConvertaBuffer speeds the process of outputting a large amount of data to the printer by receiving the data as fast as the PC can send it—1,000 characters per second, storing up to 64K bytes of data in its built-in memory and then sending it to the printer at the printer's slower speed.

The suggested price is $499.

von Leivendyke Enterprises
Silvermine Avenue
Norwalk, CT 06850
203-846-4973

CIRCLE NO. 381 ON READER SERVICE CARD

PolyLibrarian™ from Polytron

An Object Module Librarian for the IBM PC, PolyLibrarian is now available as a productivity enhancement software package for programmers using the MS DOS operating system. PolyLibrarian organizes related object code modules (.OBJ files) into a single library (.LIB file). The IBM PC linker can then automatically select only the modules necessary to construct an executable program (.EXE file).

Available for $99, PolyLibrarian works with any compiler or assembler that uses the IBM PC Linker.

Polytron
P.O. Box 787
Hillsboro, OR 97123
503-640-2875

CIRCLE NO. 386 ON READER SERVICE CARD

IBM and Nestar

Nestar Systems, a leading supplier of full performance networking systems for personal computers, has been named a value-added dealer (VAD) by IBM for its personal computer family. Under the agreement Nestar can sell its networks, including IBM PCs, to end-users in the United States.

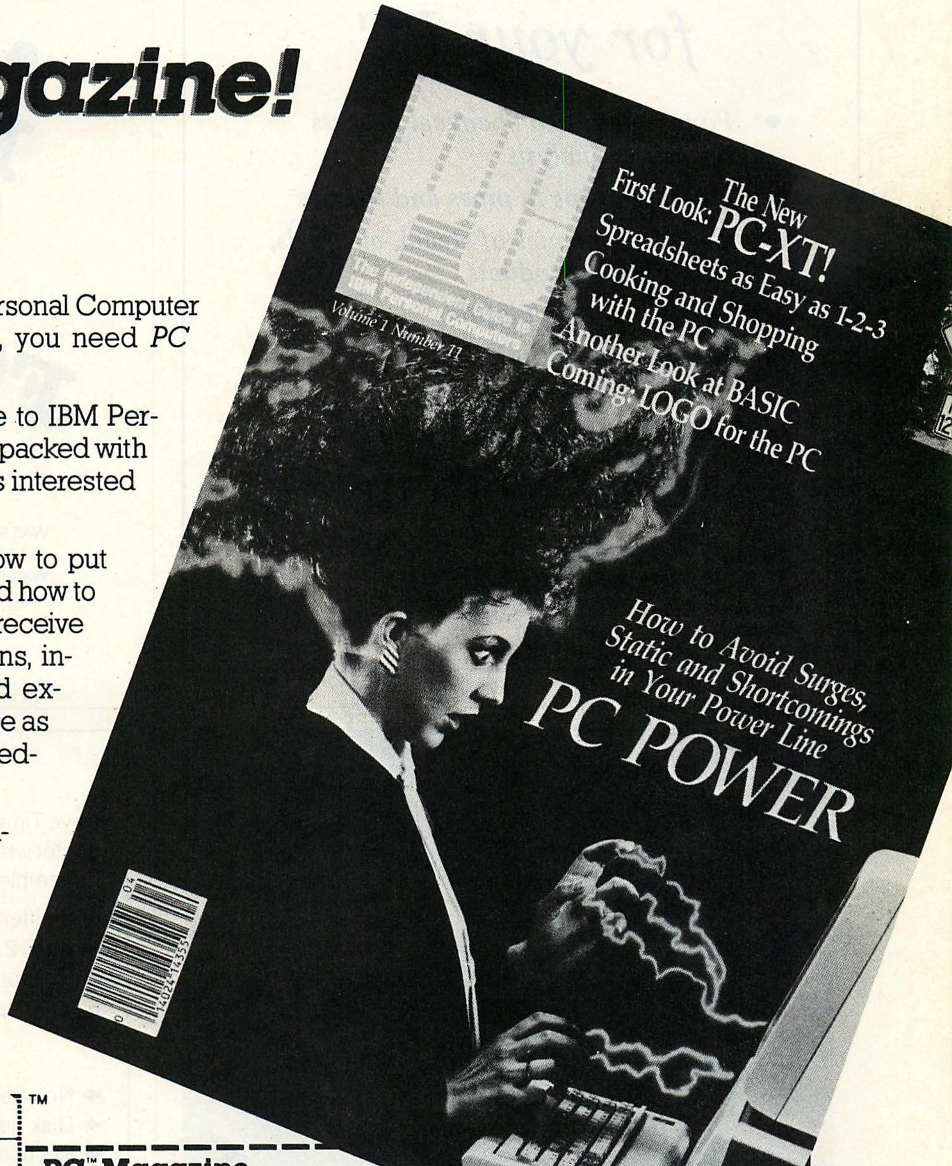
Nestar is well known for its PLAN 4000, the basis for its concept of Community Microcomputing™, computer usage that combines dedicated, accessible personal computing power with full communications capability for sharing and accessing information.

Nestar Systems, Inc.
2585 East Bayshore Road
Palo Alto, CA 94303
415-493-2223

CIRCLE NO. 385 ON READER SERVICE CARD

Get all the facts about the IBM Personal Computer.

Get PC™ Magazine!



Whether you already own an IBM Personal Computer or you're thinking about buying one, you need *PC* Magazine.

PC Magazine is the independent guide to IBM Personal Computers. Each monthly issue is packed with the latest information for everyone who's interested in IBM Personal Computers.

This is *the* magazine for finding out how to put together the best IBM "PC" system... and how to get the most out of it. Every month you'll receive hundreds of colorful pages of evaluations, insights, and straight talk from respected experts—professionals in computer science as well as lawyers, businessmen, writers, educators and many others.

PC covers software, hardware, applications, and every other topic of importance to the thousands of IBM Personal Computer users who read it. To make sure that we give you the facts you need, we include a special "User-to-User" section, a "PC Wish List," and news about IBM Personal Computer clubs, events and publications.

Right now you can save up to 31% on an introductory subscription. And if you're ever dissatisfied with *PC*, just write and tell us—you'll receive a prompt refund for all the unmailed issues remaining in your subscription.

PC™ Magazine
P.O. Box 598, Morris Plains, NJ 07950

Yes! I want to subscribe to *PC*, the independent guide to IBM Personal Computers:

- ☐ One year (12 issues) only $21.97–19% off!
- ☐ Two years only $40.97–24% off!
- ☐ Three years only $55.97–31% off!

8H344

Savings based on full one-year subscription rate of $26.97.

CHECK ONE: ☐ Payment enclosed. ☐ Bill me later.

Mr./Mrs./Ms.____________________
(please print full name)

Company____________________

Address____________________

City________ State________ Zip________

Offer valid only in the U.S., its territories and possessions. Please allow 30 to 60 days for delivery of first issue.

Engineering Software

for your PC

- *Power plant and chemical process flowsheet analysis*
- *Pressure drops in pipes and fittings*
- *Thermodynamics of steam, air, fuels, combustion products*
- *Component simulation*
- *Solvers and optimizers*

Call or write for our free catalog

PSI/Systems

Research Park, P.O. Box 3100
Andover, MA 01810
(617) 475-9030

CIRCLE NO. 167 ON READER SERVICE CARD

WATSOFT**PRODUCTS**INC

A subsidiary of the University of Waterloo

Announces

a full screen editor and a family of language interpreters for the IBM Personal Computer

APL
BASIC
COBOL
EDITOR
FORTRAN
PASCAL

For further information contact:

WATSOFT Products Inc.
158 University Ave.
Waterloo, Ontario
N2L 3E9

(519) 886-3700
Telex No.: 06-955458

CIRCLE NO. 218 ON READER SERVICE CARD

Need a Transfer?

USE PX-C, PC-DOS to CP/M-86.

Available At Your Local IBM PC-Digital Research Dealer

INTERNATIONAL DISK

7246 Remmet Ave., Ste. 101
Canoga Park, CA 91303
(213) 992-0514 (213) 704-6518

CIRCLE NO. 255 ON READER SERVICE CARD

DOS 1.1 and 2.0 UTILITIES
SOFTWARE PRINTER SPOOLER

Save Time!! Spooler allows the computer to be printing text to the printer while running a program. Supports Visicalc, Basic, Macro-Assembler, Multiplan, FinalWord, Wordstar etc.

- 5 different buffer sizes from 4k to 64k (DOS 1.1)
- DOS 2.0 buffer sizes to 320 kbytes
- Supports Xon-Xoff protocol for serial printers.

$45.00 Req. 1 drive, 64 kbytes, parallel or serial printer

RAM DISK

- No internal switches to set on the PC
- Disk sizes from 32k to 320 kbytes (32 kbyte steps)
- Keyboard reset (Ctl-Alt-Del) is disabled so you won't lose data
- Click from PC speaker can be enabled for disk access signal

Price $40.00 Requires 96 kbytes and 1 disk drive
Ram Disk and Software Printer Spooler both for $69.00

COPYLOCK PROTECTION PROGRAM

An effective protection system for your .COM or .EXE file which is not copyable by any of the general copy programs such as Copy-PC, COPY2PC, System Backup v1.3 and the DOS Disk-copy command. Disk can also be made a "bootable" disk by the end user by using the SYS command from DOS.

- Supports Compiled BASIC programs with chaining

Requires 2 drives, 64 kbytes. $799.00

See us at PC'83 EAST
Bayside Exposition Center
Boston
Saturday-Monday
October 8-10 1983

SoftDesign

617 861-0902
6 Blodgett Rd. Lexington, Ma. 02173

CIRCLE NO. 198 ON READER SERVICE CARD

Whitesmiths Ltd.
Fighting the Pirates of Software

Since the beginning of time, man has searched for the answer to two questions:

what is the meaning of life and how can I keep pirates from ripping off my software?

Max Stul Oppenheimer

The latter question about software piracy is far from being answered, and it pits software writers against users in unavoidable conflict. The conflict is simple to state: a careful user probably wants to know how the program he or she is relying on works, probably wants to keep back-up copies of the program, and probably wants the option to use the program on more than one machine (at least on a temporary basis if the primary machine is being serviced); a software writer's desire to maximize sales will motivate him to restrict the user's access to his or her programming techniques and restrict the use that may be made of any given copy of the program. When a large enough custom project is involved, a sophisticated user and a sophisticated software writer can bargain and reach a compromise each can tolerate. Off-the-shelf software does not lend itself to bargaining—the price usually will not support the effort involved, and often at least one party is not experienced enough to know what is at stake.

Software houses have turned to a rather limited variety of approaches to deal with the conflict: trade secrecy (*e.g.*, copy protection), copyright, patent, or a hybrid of the foregoing. Each approach has drawbacks. Patents are expensive and difficult to secure, copy protection schemes devised to date have had very short useful lives, copyright is somewhat esoteric, and enforcement of copyright requires legal proceedings as opposed to the self-executing nature of effective copy protection.

For certain of its software, Whitesmiths Ltd. has opted for what it refers to as "licensing under copyright." The evidence of the license is an "authorization seal," which the user acquires along with the software. The user must affix the seal to the system on which the software is to be used. Whitesmiths solicits third-party reports of the use of its software on machines not bearing the appropriate seal and presumably would investigate such reports and take legal action to enforce its copyrights as warranted. Whitesmiths claims that this eliminates the need to negotiate license agreements and will reduce the costs of its software.

Whitesmiths misplaces its emphasis on the talisman of the seal; the claim that the seal eliminates the need for a license agreement is simply wrong. It eliminates a formal written agreement that must be signed by the user, but the user's acceptance of the software/seal package may constitute an agreement that binds the user to Whitesmiths' terms. Whether the authorization seal mechanism effectively creates a contract between Whitesmiths and the user depends on

Max Stul Oppenheimer is a partner in a large Baltimore law firm, Venable, Baetjer and Howard.

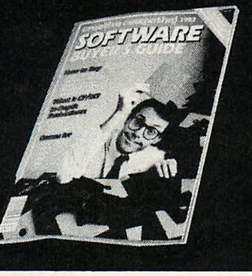
whether the user has been informed of the terms of the contract and has the legal intent and capacity to enter the contract. Although perhaps not a problem in the majority of Whitesmiths sales, any software writer who attempts to impose a contract rather than negotiate one (whether by seal or by a licensing agreement shrink wrapped with the software media) runs the risk that the purchaser is a minor who may not be bound by the terms of the license.

Assuming that the user does, by the act of purchasing the software and accepting the authorization seal, become bound by the licensing agreement, the question is what the terms are of that agreement. There is no written licensing agreement, but the user can look to Whitesmiths' press releases and catalog to attempt to determine what Whitesmiths might claim to be the terms of the license agreement.

The press releases indicate that Whitesmiths has made significant compromises: the software can be stored on disk or Winchester drive, and users are permitted to make archival copies for their own use. However, the press releases also indicate that Whitesmiths intends the software to be used only on a single computer bearing the appropriate authorization seal. Taken literally, this would mean that if a user's machine were down for maintenance, the software could not be run on a back-up machine. It might be logical to assume that under such circumstances, Whitesmiths could not reasonably have any objection to removing the seal from the down machine and placing it on the machine in use. However, in its March 30, 1983, news release, the company states that its software "can be bought and sold along with the computer for which it is authorized." It also states that a customer "may not buy one copy of our software and circulate it among friends."

Taken together, this suggests that Whitesmiths intends to place

How to find the right software for your home or office computer.



The 1983 Software Buyer's Guide saves you time, trouble and money when you're shopping for software for your personal or business computer.

In this convenient, comprehensive guide the editors of **Creative Computing** have compiled up-to-date software reviews from our magazine. It's the one and only reference you need to select the right software for your individual computer needs.

Included in this new edition are programs for every conceivable application, including practical programs like TK! Solver, Vision, Paint and Video Easel and Slide Show–plus the latest games for Color Computer, TRS-80, Apple, Atari, and more!

You'll also find step-by-step guidance on buying a word processor, database manager, educational or recreational software.

And once you decide on the programs you want, you can find out where to get them though the convenient directory of manufacturers that's included.

Get the hard facts about software in the **1983 Software Buyer's Guide.** Send for your copy today. Only $4.95!

MAIL TODAY TO:
Software Buyer's Guide
CN1914
Morristown, N.J. 07960

Please send me_____copies of the **1983 Software Buyer's Guide** for only $4.95 each, ($3.95 plus $1.00 per copy postage & handling). $6.00 outside U.S.A.

Total Amount $__________

☐ Payment Enclosed (CA, NJ and NY State residents please add applicable sales tax.)

NAME__________________
(please print)

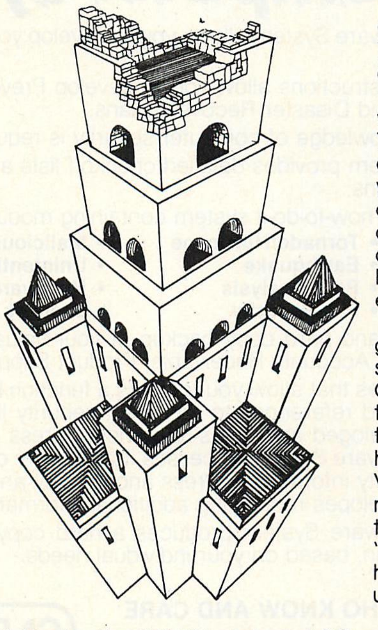
ADDRESS__________________

CITY__________________

STATE________ZIP________

Direct Disk-to-Disk File Transfer with Xeno-Copy PLUS™

THE TOWER OF BABEL CONQUERED!



In the Old Testament, man's excessive ambition results in the Tower of Babel where no one speaks the same language. In Modern Times, the microcomputer industry has achieved the same result without any direct divine intervention. Now, with **Xeno-Copy PLUS**, you can read AND write over 35 different disk formats in your IBM PC or PC-compatible.

Place your "foreign" disk in your PC. READ and WRITE files between PC-DOS and your "foreign" disk. No modems, serial links, or other hardware required. Capabilities include single and double-sided formats, CP/M-80, CP/M-86, TRSDOS, p-SYSTEM, and others.

The fast and easy way to move disk files TO and FROM your other computer, using your PC as the host computer. Your other computer is not required, just the disk. Fully menu-driven and easy to use.

Xeno-Copy PLUS gets you the best in user support. Your Xeno-Copy package contains our toll-free (800) technical support hotline number. All Vertex registered users receive a free subscription to our newsletter "XENOtes" containing helpful info and upgrade notices.

Xeno-Copy PLUS available for the IBM PC and most PC-compatibles. Become a Xeno disciple for only **$124.50.**

THE PC KEYBOARD . . . CONQUERED

Smite thine keyboard limitations with **KEYSWAPPER 2.0's** punch! First, customize your keyboard. Use four different built-in keyboard layouts including DVORAK (ANSI). Move any key. Use numeric keypad and cursor controls at the same time. Hear audible feedback on CAPSLOCK/NUMLOCK, and reset CAPSLOCK like typewriter. Menu-driven for easy setup.

Then, add the most powerful macro-command capability around. Program ANY key as a macro-even relocated keys or multiple keys. Most macro features can be changed even while another program is running: define a new macro, toggle any **KEYSWAPPER** feature on/off including macros, bring up a macro definition table and "Help screen". Macros can contain user input and time delays. Create "phantom" menus transparent to your application program. Save macros in printable, editable files. Optional date/time at cursor prompt, disable selected keys, and other advanced features.

Works for you on the PC, XT and selected PC-compatibles, under DOS 1.1 and 2.0. Compatible with most popular programs. Only 69 sheckels ($69.95 U.S. currency).

THE *ORIGINAL* KEYFIXER™

An original gift from the three wise men? Probably not, but these precision molded collars fit onto the existing ENTER, TAB, BACKSPACE, and both SHIFT KEYS to increase accuracy and reduce fatigue. Unique design does not change key height. Imprinted with "ENTER" as in IBM manuals. Original equipment grey color matches existings keys. A thousand and one satisfied customers.

Set of FIVE .... only $14.95

Check with your local retailer, or call us for your local Vertex dealer.

Vertex systems

**Dept M, 7950 W. Fourth St.,
Los Angeles, CA 90048
(213) 938-0857**

If unavailable locally, send payment or VISA/MC with card # and exp. date. Phone orders accepted. Add $2.00 ship/handling. CA residents add 6½% sales tax.

CIRCLE NO. 260 ON READER SERVICE CARD

restrictions on a user's disposition of its software beyond those afforded by statutory copyright. There would, therefore, be at least an issue as to whether an owner could temporarily transfer the seal among his own machines. Were a user negotiating the license with Whitesmiths, his argument would be that these restrictions are akin to selling a book for use in the living room only and that, under copyright law, the sale of a copy by the author gives the buyer the right to dispose of that copy as he pleases.

Section 101 of the 1976 Copyright Act specifically permits the owner of a lawful copy of a copyrighted work to sell "or otherwise dispose of the possession of that copy" without obtaining permission from the copyright owner. The report of the congressional committee prepared in connection with the 1976 Copyright Act offers the following example: ". . . for example, the outright sale of an authorized copy of a book frees it from any copyright control over its resale price or other conditions of its future disposition. A library that has acquired ownership of a copy is entitled to lend it under any conditions it chooses . . . ." Note that both the statute and the committee report are focusing solely on the rights of the owner of a copy; were Whitesmiths leasing rather than selling its software (as IBM purports to do in its licensing agreement), the Copyright Act would not provide explicit permission to make such transfers without the copyright owner's permission.

In addition to the explicit statutory authorization of disposition of copyrighted works by the owner of the copy, there is a legal doctrine known as "fair use," which permits certain uses of copyrighted works without the permission of the copyright owner. Greatly oversimplified, the doctrine permits very limited uses (an easy example is quotation of limited portions of a copyrighted work in connection with a review of that work) where very limited harm will be done to

the economic value of the copyright. Query (legalese for your guess is as good as mine but neither is as good as the judge's) whether use by the same user on multiple machines would be viewed as fair use.

Has Whitesmiths explained the meaning of life? The use of an authorization seal may make detection of unauthorized use easier if, as Whitesmiths solicits, third parties report observations of the use of Whitesmiths software on machines not bearing authorization seals. It is

Certainly one of the main problems in battline software piracy is detection, and Whitesmiths may well have discovered a valuable weapon.

an interesting innovation. Certainly one of the main problems in batting software piracy is detection, and Whitesmiths may well have discovered a valuable weapon.

However, the use of the authorization seal does not eliminate the licensing agreement—it simply eliminates the negotiation of the terms of the agreement. Whitesmiths has made a significant compromise in explicitly permitting the user to make archival copies. Whitesmiths has also imposed terms beyond what statutory copyright would require. To the extent that the licensing agreement goes beyond the copyright statute, any violation would be a breach of contract but not a violation of copyright law. The distinction can be significant in terms of the remedies available to Whitesmiths and the defenses available to the user.

Copyright law does not prohibit the copyright owner from asking more, by way of contract, than the statute grants him. Users should be aware that the copyright law also does not prohibit the owner of the copyright from giving up rights that he otherwise would have under the statute. ■

TECH BOOK

A Special Section for Product and Service Listings

ACCESSORIES/SUPPLIES

DISCSAVERS™

Multi-Color vinyl sleeves for 5¼ and 8-inch diskettes that provide a 4-color filing system, durability, ease of labeling and secure fit. Packaged 12 or 10 (8") per box, they are available in red, blue, yellow and green. Ideal for home or office use.
(List Price: $6.29 for 5¼", $8.75 for 8")

ROCKROY, INC.
COMPUTER PRODUCTS DIVISION
7741 East Gray Road
Scottsdale, AZ 85260
(800) 528-2361, (602) 998-1577

IBM PC SECURITY

Deter unauthorized persons from opening PC system unit. Kit includes specially designed Clarke screwdriver & companion Kent screws, kit price $18.95. Control multiple PC'S..qty disc & extra screws available. To order send ck or money order:

HOOKER TECH/ETERIA
PO Box 461
Lawrence, NY 11559

COMPUTER SERVICES

INSURANCE

SAFEWARE, unique insurance for microcomputer owners, covers all hardware, media, purchased software. SAFEWARE provides full replacement coverage, after a low $50 deductible, for as little as $35 a year. Protect your system against power surges, fire, theft, accidental damage. For immediate coverage call toll-free 1-800-848-3469. In Ohio, 1-800-848-2112.

COLUMBIA NATIONAL GENERAL AGENCY
88 E. Broad
Columbus, OH 43215

HARDWARE/ADD-ON BOARDS

QUADCOLOR VIDEO CARDS

Quadcolor, designed for the IBM PC, has color bit-mapped graphics for sixteen colors on the screen at once. Quadcolor I is functionally equivalent to IBM's card and retails for $295. Quadcolor II offers 640 x 200 high-resolution for $275. And Quadcolor III offers 640 x 400 high-resolution.

QUADRAM CORP.
An Intelligent Systems Company
4355 International Blvd.
Norcross, GA 30093
(404) 923-6666
TWX 810-923-6666

FLEXIMEM ROM/RAM CARD

For resident virtual programs, this is the ONLY board to buy! Use as read/only (ROM) board for faster access to programs or expand read/write (RAM) memory up to 128K. Compatible RAM and ROM/PROM/EPROM/EEPROM devices can be installed on same board for maximum flexibility. Call for technical note 47A.

RAYTRONICS
4901 Morena Blvd., Building 900-C
San Diego, CA 92117
(800) 854-1085, (619) 270-4000 (CA)

PC ADD-ONS

We carry everything from memory upgrades and disk drives to memory cards and hard disks. Brands include Tandon, CDC, Seattle Computer, Tecmar, Tallgrass, and more. U.S. Inquiries: For service and satisfaction call:

EMJ DATA SYSTEMS LTD.
79 Regal Rd #5
Guelph, Ontario, N 1K 1B6 Canada
(519) 837-2444, (416) 846-3233

256K SLOTLESS MEMORY

Let us install up to 256K RAM on your EXISTING SYSTEM BOARD using NO SLOTS. We improve your IBM PC so it can address the 64K RAM chip 128K—$192, 192K—$256, 256K—$320 (expandable and warranted). DO-IT-YOURSELF KITS—PC-KNB—$39.95, PC-KPC—$59.95. 64K-PUR—$5.95 each.

Add-MEM
22151 Redwood Rd.
Castro Valley, CA 94546
(415) 886-5443

13 FUNCTION EASIBOARD™

IBM quality plus Easitech Excellence exclusively at IBM PC dealers worldwide. Memory (64K-256K), spooler, printer and modem connections, three electronic disks, chronograph and more. No need to search for low prices. Our lowest prices are at authorized IBM PC dealers only. $325 to $595. EASIBOARD by EASITECH.

EASITECH CORP.
2215 Perimeter Park, Suite 22
Atlanta, GA 30341
(404) 452-7576

THE ENCRYPTOR

THE ENCRYPTOR, a data encryption board, is designed for use with the IBM, APPLE, S100 & IBM compatible micros. THE ENCRYPTOR puts the strength of DES at the disposal of business & personal computer users. Sensitive data can be quickly & economically protected whether for storage or transmission.

FUTUREX
9700 Fair Oaks Blvd.
Fair Oaks, CA 95628
(916) 966-6836

HARDWARE/DISK DRIVES

8" & 5¼" FLOPPY DRIVES AND DISKETTES

• 8" flexible disk subsystem—Attractive thin-line design stacks above or below system's unit (19½" x 18" x 3½") ANT with complete documentation. Controller board and software available.
I-8480 DD, DS, 2.4 MB $1495
I-8481 SD, DS, 1.2 MB $995
• 5¼" internal disk drives—TANDON or Control Data—DS, 40 ATP1, 320 KB.
TM 100-2 $235 • CDC 9409 $249
5¼" half height internal drives available.
• Control Data diskettes—1240-00 5¼" SS/DD w/write protect notch in hub ring—bx of 10 $22. 1244-00 5¼" DS/DD $35. 1225-00 8" DS/DD wpn $39.50. VISA/MC.

MICROXPRESS
305 S. State College, Suite 135
Anaheim, CA 92806
(714) 632-8512

HARDWARE/MONITORS

QUADSCREEN

This 17-inch high-resolution monochrome monitor displays 10,240 characters at once with 960 x 512 dots resolution—more than five times that of IBM's PC monitor. Bit-mapped graphics allows addressability of each dot. Full IBM MSDOS/BIOS compatibility. Quadscreen comes with video card, cable and software. Retails for $1995.

QUADRAM CORP.
An Intelligent Systems company
4355 International Blvd.
Norcross, GA 30093
(404) 923-6666
TWX 810-923-6666

QUADCHROME

The RGB 12-inch color monitor has a special NEC .31mm dot pitch tube to deliver up to 690 x 480 dots resolution for a sharp screen image. Color graphics, word processing, and accounting are just a few tasks this monitor can perform. Retails from $795.

QUADRAM CORP.
An Intelligent Systems company
4355 International Blvd.
Norcross, GA 30093
(404) 923-6666
TWX 810-923-6666

HARDWARE/PERIPHERALS

VIDEOSLIDE

Get high-quality 35mm color slides for the cost of film and processing—about $.50 each. Videoslide 35 Computer Graphics Camera plugs directly into 9-pin plug on IBM PC color/graphics adaptor. Compatible with RS-170 terminals. Price of $2799 includes TTL RGB interface and all necessary cabling.

LANG SYSTEMS INC.
1010 O'Brien Dr.
Menlo Park, CA 94025
(415) 328-5555

HARDWARE/STORAGE DEVICES

WINCHESTER SUBSYSTEMS

Tallgrass Technologies offers a family of Winchester HardFiles and streaming tape by file backup for the IBM PC. With formatted capacities from 6.25MB to 70MB, Tallgrass has a HardFile System to fit the most stringent of data processing applications. From $2995 including integral tape backup.

STEVE VOLK/V.P. MARKETING
Tallgrass Technologies Corporation
11667 West 90th
Overland Park, KS 66214
(913) 492-6002

RATES AND INFORMATION

Listings are grouped by category and consist of a bold lead line (23 characters maximum), 7 lines of ad copy (45 characters per line), plus 4 lines of company name, address and telephone number.

Listings are available only on a 3 issue basis at $60. per issue ($180. total). Copy will have a set format and remain the same for all 3 months. Enhance the appearance of your ad by including your Logo at an additional cost of $25. per issue ($75. minimum extra charge). Pre-payment is required by check, money order, or American Express, Diners Club, MasterCard, Visa credit cards. Closing Date: 1st of 2nd month preceding cover date.

Send copy and remittance to PC TECH JOURNAL, TECH BOOK, 12th FLOOR, 1 PARK AVENUE, NEW YORK, NY 10016. Call (212) 725-4215 for additional information or assistance.

TECH BOOK

A Special Section for Product and Service Listings

Mailing Lists

PROFILES OF RETAIL STORES & USERS

CI can provide names & addresses of: a) over 2,200 retail computer stores surveyed via the telephone, b) new personal computer users (i.e. IBM, Apple, Osborne, etc.). Both can be tailored & used for mailing lists, telemarketing or direct sales. Call (619) 450-1667.

COMPUTER INTELLIGENCE CORPORATION
3344 North Torrey Pine Ct.
La Jolla, CA 92037

MICROS IN LARGE ACCOUNTS

CI's new information service locates large-volume users of personal computers in key accounts. This program provides installed user data, network configurations, and their plans for new purchases. The data is gathered via 1500 telephone surveys each month with large companies including the Fortune 1000. Call 619-450-1667.

COMPUTER INTELLIGENCE CORP.
3344 North Torrey Pine Ct.
La Jolla, CA 92037.
(619) 450-1667

IBM MAILING LISTS

Over 23,000 names of IBM personal computer owners (counts increase daily) available for rental on labels or magnetic tape. Total 411,000 including other brands.

IRV BRECHNER
TARGETED MARKETING, INC.
Box 453
Livingston, NJ 07039
(201) 731-4382

Software/Accounting

CUSTOM PAYROLL

All systems support multiple pay categories, calculation of all taxes, user-defined deductions, exceptional item handling, a variety of necessary reports, paychecks and W-2 forms. Written in PC Basic with full source included for easy maintenance. Prices start at $400.00 for basic system and manual.

DATASMITH, INC.
Box 8036
Shawnee Mission, KS 66208
(913) 381-9118

SOLOMON III GENERAL LEDGER

Suitable for large and small businesses, banks and accounting firms. Unsurpassed in flexibility, power and ease of use. Graphically displayed screens, user defined chart of accounts, reports, journal types and budgets. Consolidation of multi-divisions (companies), auto-reversing and recurring transactions, ratio analysis plus a complete report generator. $595 complete. Demo available.

Solomon Series™ Software
Information For Wise Business Decisions

COMPUTECH GROUP INCORPORATED
Lee Blvd., Box 271
Frazer, PA 19355
(215) SOLOMON or 644-3344

Software/Business Project Management

PROMAC/DOS FOR MANAGERS

Project planning, management and control. Uses critical path method. Offers interactive control of project activities, schedules, costs and resources. Up to 10,000 activities in any work breakdown structure. Tabular and graphic reports support earned value analysis. Call or write for brochure and sample reports.

PRIMAVERA SYSTEMS, INC.
29 Bala Avenue, Suite 224
Bala Cynwyd, PA 19004
(215) 667-8600

Software/Compiler

C-SYSTEMS ● COMPILER ● WINDOW

C-Systems V7 compatible nonfloat c compiler for real-time system applications. Optimized assembler source code. 128K, 2 dd. $195. c-window is a source level debugger for c programs. Single step, breakpoint, trace at statement level. Display and change variables using full c expression syntax. Requires c-compiler $195.

C-SYSTEMS
P.O. Box 3253
Fullerton, CA 92634
(714) 637-5362

C86—"C" FOR PROFESSIONALS

"C" brings great power, control and conciseness to programming. C86 brings FULL UNIX syntax (portability), source to a rich library (gives you control), no royalties, PCDOS & CPM-86 support, and overlays. Ask a friend—we are ahead. Development will keep us ahead. Hotline support is solid. $395. Call your Dealer or CI.

COMPUTER INNOVATIONS, INC.
10 Mechanic Street, Suite J-113
Redbank, NJ 07701
(201) 530-0995

THE PROGRAMMER'S SOURCE

COMPARE products. Find what's best. We serve programmers of MSDOS and CPM systems. We cover all compilers (like 15 PCDOS C compilers), editors, utilities, operating systems, communications programs, addon hardware, key books and supplies. Call for answers, Referral Service, a catalog, literature, prices.

THE PROGRAMMER'S SHOP
135 Main Street, Suite 3-D
Maynard, MA 01754
(800) 421-8006, (617) 897-4750

Software/Data Management

T.I.M. III

A data base management system that generates and summarizes files, mailing lists, and employee records. The data can be accessed and searched in a number of ways. T.I.M. provides help screens and function key legends as user aids. Capacities include 32,767 records per file, 40 fields per record, 60 char/field. (List Price: $500.)

INNOVATIVE SOFTWARE, INC.
9300 W. 110th St. #380
Overland Park, KS 66210
(913) 888-0154

Software/Development Tools

METAFILE DEVELOPMENT SYSTEM

The METAFILE Software System creates a development and operational environment that provides integrated facilities for high level programming, word processing, data base management, modeling, report generation, communications, forms generation, menu building, data conversion, and impromptu interaction.

METAFILE

SENSOR-BASED SYSTEMS
Olmstead Federal Building
Chatfield, MN 55923
(507) 867-4440

BASIC DEVELOPMENT SYSTEM

An integrated set of software tools that will greatly increase the efficiency and productivity of programmers using PC BASIC. Includes Single Step Trace, complete cross-reference, dynamic dump of all program variables, renumber with program line relocation, program compression and expansion. $79, VISA and MC accepted.

SOFTOOL SYSTEMS
8972 E. Hampden Ave., Suite 179
Denver, CO 80231
(303) 793-0145

PASCAL BIOSLIB

Give Microsoft Pascal all the power of the PC's BIOS roms. Graphics, color, cursor control, scrolling supported. Use all keys including ALT and all 40 Function Keys. Use communications adapters, multiple printers. Determine installed equipment and memory. $39.95 for diskette and documentation.

NEXT GENERATION SOFTWARE
144-10 Roosevelt Avenue
Flushing, New York 11354
(212) 762-2833

ACTIVE TRACE

Automates the tedious part of PC BASIC programming/debugging, leaving you time to learn & create. Pops a window into BASIC which displays variable values as your program runs. Output to screen, printer, or disk. No need to single step. Complete XREF mapping. Review in BYTE 4/83 p.344. $79.95 CALL FOR INFO.

AWARECO
P.O. Box 695
Gualala, CA 95445
(800) 358-9120, in CA (707) 884-4019

TECH BOOK

A Special Section for Product and Service Listings

LANGUAGE ORIENTED EDITOR

The Entry System for Programs is the easiest way to program in Pascal and C. Great for expert & beginning programmers. Formats your code and checks for syntax errors while you edit. One keystroke builds complete skeletons and structures. Write error free programs 50% faster, 100% better. $245. Call for Demo Disk.

BELLESOFT
THE FIRST NAME IN PROGRAMMING PRODUCTIVITY

BELLESOFT, INC.
2127 Bellevue Way SE
Bellevue, WA 98004
(206) 451-9788

SOFTWARE/FINANCIAL

CALCULATOR EMULATOR

Now you can make quick calculations with the ease of an adding machine or print calculator on your IBM PC. Includes programmable formats for scientific notation, fixed or floating point, auto decimal entries, PLUS optional title line and labelling of entries. For home or business. Dealer inquiries welcomed. $39.95. MC/VISA/CHECK/MO.

SPECIALTY DESIGNS
7739 El Pensador
Dallas, TX 75248
(214) 960-2455

SOFTWARE/GENERAL

SAPANA PROGRAMS PLUS

MAIL-TRACK-1 (with LetterMerge) $29.95, EXPENSE-TRACK-1 $29.95, SOFT-INDEX-1 (an electronic card file) $49.95, HOUSEHOLD-TRACK-1 $49.95, INVOICE-GENERATOR-PA/DO $99.95, PAY-ROLL-TRACK-1 $99.95. Directory & Guide (free sample). IBM DISKETTES (SSDD $29.90, DSDD $39.90). ADD P&H. 30 DAY MONEY BACK GUARANTEE.

SAPANA MICRO SOFTWARE
Dept. PCT 1, 1305 S. Rouse
Pittsburg, KS 66762
(316) 231-5023

RAM FAST SORT

Sorts 500 strings (like names, dates and #'s) in 4 seconds with optimized bubble and quick sorting. Use multiple arrays (1 primary, 15 secondary), of any length. Written in machine code. Uses compiled MS Basic (8086). Comes with: high speed string search & expanded basic input routines. $34.95

GSC, INC.
Route 2, Box 466
Alachua, FL 32615
(904) 462-4564

SOFTWARE/GRAPHICS

FORTRAN GRAPHICS PACK

57 MicroSoft or SuperSoft FORTRAN callable subroutines. Professional graphics power backed by a 90 page user manual, filled with examples and designed to be helpful to every level of user. Five packages in one! General utility, 2-D interactive, 2-D plots, 3-D plots, and solid models (hidden line removal!). $95.

MICROCOMPATIBLES
11443 Oak Leaf Drive
Silver Spring, MD 20901
(301) 593-0683

CHEETAH BUSINESS GRAPHICS*

Starting at under $1000, the Cheetah Business Graphics System is upgradable to Gene-graphics-quality resolution, with a variety of intermediate resolutions and hardcopy film or plotter output options. Infinitely friendly and flexible. You can make a slide in 10 minutes for as low as $5.00. *Elected by IBM as a VAR for graphic content.

CENTEC CORPORATION
11260 Roger Bacon Drive
Reston, VA 22090
(703) 471-6300

GIRAPH BUSINESS GRAPHICS

GIRAPH: Line, bar, stack bar and pie charts. Menu-driven with selectable shading, labeling, colors, etc. Will use VisiCalc and Multiplan files ($199). GIRAPH SUPER: All the above features PLUS scatter diagrams, 3-D bars, separate left and right scales, log scales, text anywhere and more ($299). Demo $25. Call or write to:

DATA DISPLAY
171 W. 4th Street
NY, NY 10014
(212) 620-8134

VERSAWRITER

Complete drawing system for IBM PC includes digitizer drawing board and software—ready to use. Use digitizer to trace drawings to the computer screen. Features include draw & dot mode, fill & airbrush in 100 colors, add text, save and recall picture, dump graphics to printer, make circles, boxes, pie charts & bargraphs, create your own brush palette and more. $299.00.

VERSA COMPUTING, INC.
3541 Old Conejo Road, Suite 104
Newbury, CA 91320-2178
(805) 498-1956

PLOT-10 GRAPHICS EMULATOR

PC-PLOT is a complete emulation of the TEKTRONIX model 4010 graphics terminal. The basic packages range from $40 to $75 plus optional plotter driver software. Write or call for a technical brochure.

MICROPLOT SYSTEMS CO.
1897 Red Fern Drive
Columbus, OH 43229
(614) 882-4786

35mm SLIDE FROM YOUR PC

COMPUTER SLIDE EXPRESS converts graphic files produced on the IBM PC into brilliant 35mm color slides with color resolution 400% better than your monitor. Leave your printouts behind. Project yourself into the technology of tomorrow. Use high resolution color slides at your next presentation. COMPUTER SLIDE EXPRESS. $6/slide, $30 minimum order.

VISUAL HORIZONS
180 Metro Park
Rochester, NY 14623
(716) 424-5300

GRAPH 'N' CALC

Bridge the function gap between spreadsheets and graphics. A unique graphics/analysis system that uses techniques such as Seasonality Trending, Exponential Smoothing and Multiple Regression Analysis to develop projections. Then visualize the results using vivid color video or hard copy graphic presentations.

DESKTOP COMPUTER SOFTWARE, INC.
303 Potrero Street, Ste 29-303
Santa Cruz, CA 95060
(408) 458-9095

SOFTWARE/LANGUAGES

PC/FORTH

Fast, extensible, interactive language ideal for graphics, communications, and process control application. Extension packages for 8087 numeric processor support, advanced graphics, and cross-compilation also available. Runs under PC-DOS or CP/M-86 with 48K and one disk drive. Prices start at $100.

LABORATORY MICROSYSTEMS
4147 Beethoven St.
Los Angeles, CA 90066
(213) 306-7412

ADA ON A MICRO

JANUS/ADA is an exceptional programming language for software development. Being a subset of the Ada programming language JANUS/ADA implements those features of Ada which are directly needed by professional programmers. These features include separate compilation, user defined data types and helpful program debugging.

R.R. SOFTWARE
P.O. Box 1512
Madison WI 53701
(608) 244-6436

SOFTWARE/MAILING PROGRAMS

POWERIZE

Powerize your EasyWriter/Volkswriter letter-writing with these MBS software tools: Merge'n Print for automatically repeating customized form letters (keyboard and file insertions, etc.)—$100; MBS Label Program with fast, 6-way sorting—$95; and Data Base for other follow-up and client data—$225. VISA/MC.

MBS SOFTWARE
12729 N.E. Hassalo St.
Portland, OR 97230
(503) 256-0130

SOFTWARE/MARKETING RESEARCH

SURVTAB™

Easy-to-use system for entering, editing, and tabulating data from survey questionnaires. Produces cross-tabs, frequency displays, and summary statistics for up to 80 questions and as many records as your disk permits. Perfect for market research surveys! Includes 43-page manual with instructions and examples. $180.

STATISTICAL COMPUTING CONSULTANTS
9025 Andromeda Drive
Burke, VA 22015
(703) 455-2379

SOFTWARE/OPERATING SYSTEMS

OS DEVELOPMENT TOOLS

Universal development interfaces and IRMX 86 implementations for PC DOS/MS DOS based systems. A complete family of development tools. For further information contact:

RTCS

RTCS/REAL TIME COMPUTER SCIENCE CORP.
PO Box 3000-886
Camarillo, CA 93010
(805) 482-0333

Continued on page 214

INDEX TO ADVERTISERS

TECH BOOK

A Special Section for Product and Service Listings

SOFTWARE/SORT

OPT-TECH SORT/MERGE

Extremely fast machine language Sort/Merge program. Sort ascending or descending on multiple fields, merge files. Works with variable or fixed length records. On-line help, full documentation and much, much more. Best features on the market! Compare before you buy any other. Write or call for more information. $75.

OPT-TECH
P.O. Box 2167
Humble, TX 77347
(713) 454-7428

SOFTWARE/STATISTICS

FORTRAN SUBROUTINES

100 Fortran subroutines covering data screening, correlation and regression, design, discriminant, factor, and time series analysis. Nonparametric statistics, generation and distribution functions, matrix and storage operations, Eigen analysis, matrix & polynomial solutions, & misc. linear analysis. SOURCES INCLUDED. Price $295.00, CA residents, please add 6% sales tax. Check or money order.

ALPHA COMPUTER SERVICE
P.O. Box 2517
Cypress, CA 90630
(714) 894-6808
The Source: STA550

SOFTWARE/UTILITIES

THE FORMS DESIGNER™

Attention IBM Pascal, FORTRAN, and Assembly users! Save time in designing formatted screen I/O. Interactive Forms Editor allows you to draw lines and boxes, define fields, and edit text. Access forms or read keyboard entry by writing only one line of code. Provides sequential data retrieval and storage. Requires 128K RAM. Only $275 complete. Demo and manual $35. Call or write:

BIT SOFTWARE
PO Box 619
Milpitas, CA 95035
(408) 262-1054

SUPER TOOLS FOR PC

• PFAS is the easy to use, proven B+ tree ISAM designed for Pascal. It supports multiple key access and unlimited file and record size.
• PhosForm is the fill-in form input for Pascal allowing interactive design of screens. It performs data conversion validation.
PFAS-$200.00 PhosForm—$95.00

GRYPHON SYSTEMS
922 Grange Hall Rd.
Cardiff by the Sea, CA 92007
(619) 436-1455.

PC DISK ID

Eliminates the need, under PC DOS 2.0, to reformat disks in order to add, change, or delete volume labels. Prints sorted list of files (including hidden files) from disk directly to labels or to a listing. Also allows recovery of files erased inadvertently. Requires PC DOS 2.0, 1 disk. $30.

ErgoSoft
P.O. Box 454
Oakhurst, NJ 07755
(201) 493-8352

SOFTWARE PROTECT DEVICE

Matchbox-sized Software Protection Device (SPD) attaches externally & works on both PC and XT. Each SPD is custom manufactured to uniquely respond when your program checks for it. Prevent unauthorized use of your software while allowing end-users to back-up. Prices vary from $17.50 to $25/unit depending on quantity. We have a complete line of software products for the IBM and others. (i.e. PC PROTECTOR which does what PC SPACE PADLOCK & COPY PROTECTOR do for less . . . $75). Also available: RS-232C SPD & a high level disk copy protect that CANNOT be copied by ANY back-up program. Call for details on these & other products.

SAFETY FIRST
13031 San Antonio Drive
Norwalk, CA 90650
(213) 868-7820

PC FILECOMPARE

Compare disk files with meaningful results. See the difference when a non-compare occurs. Filecompare displays the actual records that failed to compare, not just the offset into the files. Also allows comparison of files of different lengths, individual files or allows specification of global file name characters. $50. NJ + 6%. MC/VISA.

ERGO-SOFT
PO Box 454
Oakhurst, NJ 07755
(201) 493-8352

COPY PROTECTION BY MSD

The ULTIMATE diskette copy protection system designed for software developers and publishers. A variety of protection methods are available to suit your protection requirements. All IBM PC and XT environments are supported which includes: DOS, BASIC (compiled and interpretive). Pascal, P-System and stand-a-lone systems. Customized systems also available. Requires: 64K, two disk drives.

MICRO-SOFTWARE DEVELOPERS, INC.
555 Festes Avenue
Schaumburg, IL 60193
(312) 980-8338

COPY PROTECTION

Protects your .COM or .EXE program from copy programs such as System Backup v1.3, Copy-PC, and all other general copy programs. User can use the disk as a normal disk. SYS command can be used to make disk bootable. Automatically loads. .EXE files. Random method precludes copying. Price $799.

SOFT DESIGN CO.
6 Blodgett Rd.
Lexington, MA 02173
(617) 861-0902

DISK UTILITY

DISKMAP/PC is a disk utility which allows you to recover erased files and display, alter or copy the contents of any sector on a diskette. Displays are in hex and ASCII, and patches can be entered in either. Displays directory including erased files. Requires DOS 1.0 or 1.1. $59, CA residents add 6.5% sales tax.

UNICORN SYSTEMS COMPANY
3807 Wilshire Boulevard
Los Angeles, CA 90010
(213) 380-6974

DISASSEMBLY CROSS-REF

The ULTIMATE research tool for the assembly language programmer. Captures output from the PC-DOS DEBUG utility and prints a disassembly listing marking each referenced address. This is followed by a sorted cross-reference of all hexadecimal byte and word values appearing in the disassembly list. $49, VISA and MC accepted.

SOFTOOL SYSTEMS
8972 E. Hampton Ave. Suite 179
Denver CO 80231
(303) 793-0145

COPY PROTECTION

SLK/F places an assembled or compiled program on a diskette with 4 different copy-resistant features in such a way that it runs normally, but cannot be copied by backup programs such as COPYPC. The rest of the diskette is available as normal, and DOS may be added. Price $150.

OLIVE
BRANCH
SOFTWARE

OLIVE BRANCH SOFTWARE
1715 Olive Street
Santa Barbara, CA 93101
(805) 569-1682

FINALLY HD'S & 2.0

MEDIA MAGICIAN! Absolute and file relative disk editing for hard/floppy disks. For any MS-DOS 2.0 computer, plus IBM 1.0/1.1/2.0 custom version. Instant HEX/ASCII display and editing to boot tracks, directories, FAT'S, any place. Advanced W/P type functions, full screen editing and active "HELP". $68.50 and $1.50 p/h for both versions plus full DOC'S.

PHOTON
SOFTWARE

PHOTON SOFTWARE
636 120th Ave. NE
Bellevue, WA 98005
(800) 426-2675

ULTIMENU FOR DOS

Take the mystery out of DOS! Our full screen system allows you to create menus with up to 36 program entries. A single keystroke activates the selected program. Eliminates the need to memorize commands or to type lengthy character strings. $24.95 (similar system for BASIC add $19.95). VISA and MASTERCARD too!

DB/DC SOFTWARE ASSOCIATES
P.O. Box 4695
Manchester, N.H. 03108
(617) 682-6629

SOFTWARE/WORD PROCESSING

VOLKSWRITER INT'L EDITION™

For IBM-PC and compatibles. On screen tutorials, full-screen editing, on-screen formatting and standard ASCII/DOS files. Permits writing in German, French, English, Spanish & Italian and generation of scientific, mathematical & engineering characters. Multiple keyboard arrangements and communications possible. $225.00.

LIFETREE SOFTWARE, INC.
411 Pacific St., Suite 315
Monterey, CA 93940
(408) 373-4718

SOFTWRITE 2.1

Softwrite 2.1 is the instant wordprocessor. Handles characters, words, and blocks. Justify (center or right) an entire page of text in seconds (or as you type). Numbers pages automatically. Variable margins, string search, underlines with any printer. 48K RAM, floppy or hard disk. $79.95. Bank cards welcome.

SOFT BYTE
P.O. Box 16506
Seattle, WA 98116
(206) 932-BYTE

USER GROUPS

PASCAL NEWS

Pascal News is the official but informal publication of the Pascal Users' Group (PUG). PUG promotes the use of, and the ideas behind, the programming language Pascal. Pascal News has seven years of back issues at present and documents source programs written in Pascal. Four issues per year—$25.

PASCAL NEWS
2903 Huntington Rd.
Cleveland, OH 44120

PC TECH JOURNAL PRODUCT INDEX

TECH CALENDAR

September

September 13-15
Federal Computer Conference
Washington, DC
Information:
P.O. Box 368
Wayland, MA 01778
617-358-5301

September 14-16
EUROMICRO 83. Ninth Symposium on Microprocessing and Microprogramming
Madrid, Spain
Information: T.H. Twente,
P.O. Box 217, 7500 AE Enschede,
The Nederlands.

September 15-18
Second Annual Twin Cities Computer Show and Software Exposition
Minneapolis Auditorium
Minneapolis, MN
800-841-7000

September 18-23
IFIP Ninth World Computer Congress
Paris, France
Information:
1815 N. Lynn St.
Arlington, VA 22209
703-558-3600

September 22-24
Second Annual Rocky Mountain Computer Show and Software Exposition
Denver Merchandise Mart
Denver, CO
800-739-2000

September 26-29
Compcon Fall 83
Arlington, VA
Information:
P.O. Box 639,
Silver Spring, MD 20901
301-589-8142

September 26-30
Conference on Networks and Electronic Office Systems
University of Reading Berkshire,
England Information:
The Conference Secretariat,
99 Gower Street, London WC1E6AZ,
England

September 29-October 1
CP/M 83
Hynes Auditorium
Boston, MA
617-739-2000

October

October 10-13
Info 83, 10th International Information Management Exposition & Conference
New York Coliseum
New York, NY 10017
212-661-8410

October 10-13
Micro 16
Dowingtown, PA
Information:
Bill Hopkins, Burroughs Corp.,
Box 235, Dowingtown,
PA 19335
215-269-1100, Ext. 332

October 17-19
Eighth Conference on Local Computer Networks, IEEE-CS Technical Committee on Computer Communications.
Minneapolis, MN
Information:
Allan I. Edwin,
Interactive Systems/3M,
220-9W, 3M Center,
St. Paul, MN 55144

October 18-21
Third Symposium on Microcomputer and Microprocessor Applications
Budapest, Hungary
Information:
I. Baba,
Scientific Society for
Telecommunication,
POB 451, H-1372 Budapest,
Hungary

October 24-26
ACM 83
New York, NY
Information:
Thomas A. D'Auria,
City of New York,
Computer Service Center,
111 8th Avenue,
New York, NY 10011
212-620-5055

November

November 17-19
Fifth Annual Northeast Computer Show and Software Exposition
Hynes Auditorium
Boston, MA
800-841-7000

Notice of conferences, workshops and calls for papers should be submitted to the TECH JOURNAL as far in advance as possible for publication in the calendar. Send submissions to PC TECH JOURNAL, Suite 211, The World Trade Center, Baltimore, MD 21202.

PCnet is the first local area network designed specifically for the IBM Personal Computer. It consists of the necessary hardware and software to set up a powerful, yet cost effective local area network for the IBM PC. While others make claims, PCnet is a local area network proven to be successful by hundreds of installations in IBM Personal Computers.

AST RESEARCH INC.
2372 Morse Avenue, Irvine, CA 92714
Telephone (714) 540-1333
Call or write for immediate response.

AST & PCnet Hardware:
- CSMA/CD LAN technique
- 7000 feet max. distance
- Low cost CATV Coaxial cable as medium for transmission
- 1 Megabit per second transmission
- Optional Mainframe connection (e.g., AST-SNA; AST-5251; AST-3780 etc.)

AST & PCnet Software:
- Disk Sharing
  Unlike other networks, PCnet does not demand expansive file server or proprietary hard disk systems for disk sharing. Almost any PC-DOS compatible hard disk including IBM's can be used for disk sharing
- File locking
  Provides locking to common data files so that common files are only accessed by one user at a time
- PC Sharing (Multi-tasking) Allows user on 1 PC to run command on another PC in the same network as if the command had been entered on that PC's keyboard
- Printer Sharing
  Allows PCs to share printers installed in other PCs.

AST & PCnet Future Products:
- Universal File Server
- Gateway

Other AST Products
- I/O Plus II
- MegaPlus
- ComboPlus
- PlusModem
- Etc....